Ziptrack Manual

TM 1200
2311.00
JULY 1983

A. ITO, W. BOSWORTH, J. RUTHERFORD,
A. LYNCH, L. TUNG, W. YANG

TABLE OF CONTENTS

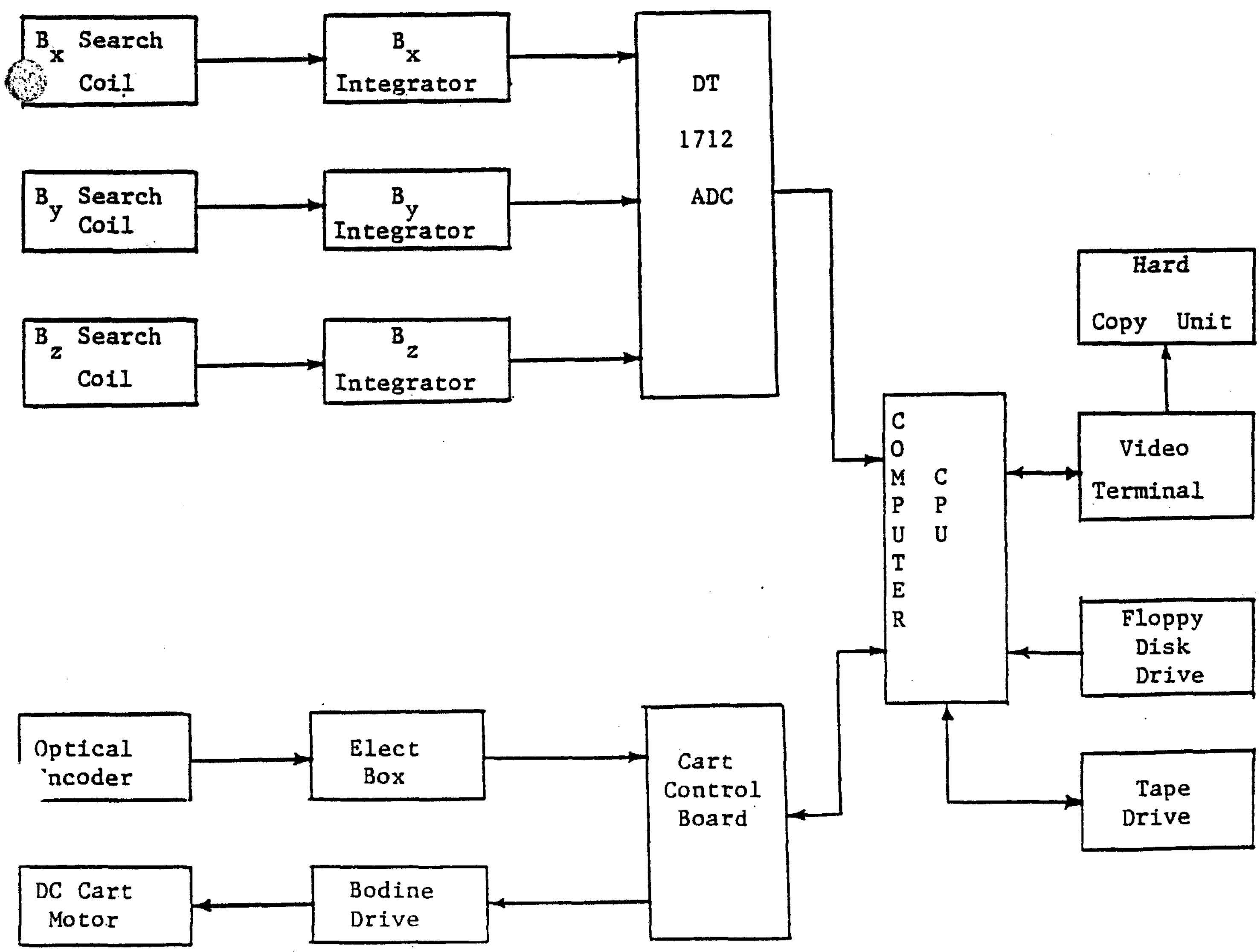
ACKNOWLEDGMENTS

We are grateful for the excellent design work done on this project by Jim Krebs and especially the late Lee Mapalo, and for the great care in the maintenance of the Ziptrack System put in by Bob Bennett and his group. We would also like to express our appreciation for the untiring efforts of Nancy Patterson and Terry Gutierrez, who helped us with the tedious job of typing and retyping.

ZIPTRACK INTRODUCTION AND OVERVIEW

GENERAL -

A phenolic cart holding 3 mutually perpendicular search coils is moved through an aluminum beam into the magnetic field. Coil voltages are integrated and digitized through a Data Translation 1712 ADC. The cart is controlled and the data is processed by an on-line PDP-11/05 computer. The results are displayed on a Tektronix 4010 terminal, stored on 9 track, 800 bpi tapes, and optionally recorded on a hard copy. Software on a floppy disk controls the system.



The position of the cart is located by an encoder and is checked along the beam line by optical switches. One encoder count equals 0.01945". The X and Y positions are changed by manipulators at each end of the beam. 625 horizontal or 500 vertical counts equal 1".

The desired field mapping can be automatically set up by programming a grid of encoder counts on the "SHOW STATUS" chart. A ziptrack command summary follows. Following that is a typical procedure for ziptrack operations. Also attached are time constants for the integrators and coil calibrations for 30 ft. and 100 ft. long cables.

ZIPTRACK COMMAND SUMMARY

CAL

Allows the user to calibrate (zero) the X, Y, and Z integrators. (See on-line examples for technique involved).

DATE

Displays at the terminal, the current date.

DISP

Displays a specified graph or chart.

DISP HELP

Lists the possible arguments to the DISPLAY command.

DISP HISTnn

Displays histogram nn where nn is a number of 01 to 99.

DISP GRPHnn

Displays graph nn.

DISP OEDn

Displays "one-event display" n.

EXIT

Exits from the program ZIPTRK. Typing CTRL C will not do anything. Any other method of exiting (such as a fatal program error), will require rebooting the system, since the console terminal and clock interrupt vectors will be corrupted.

HELP

Lists all acceptable commands on the terminal.

MOVE

Allows the user to "Manually" move the ziptrack cart and manipulators.

MOVE CART TO n

Moves ziptrack cart to encoder count position n where n is a positive or negative integer.

MOVE CART BY n

Moves ziptrack cart incrementally by n encoder counts. This too can be positive or negative.

MOVE CART TO HOME

Moves the ziptrack cart to the "home" position.

MOVE XMAN{IPULATOR} TO n (MOVE XMAN TO 73)

Moves both X manipulators to encoder count position n.

MOVE XMAN{IPULATOR} BY -n,

Moves both X manipulators by -n encoder counts.

MOVE YMAN{IPULATOR} TO -n

MOVE YMAN{IPULATOR} BY n

MOVE NEAR XMAN{IPULATOR} TO n

MOVE NEAR XMAN{IPULATOR} BY n

MOVE NEAR YMAN{IPULATOR} TO -n

MOVE NEAR YMAN{IPULATOR} BY -n

MOVE FAR XMAN{IPULATOR} TO -n

MOVE FAR XMAN{IPULATOR} BY n

MOVE FAR XMAN{IPULATOR} TO n

MOVE FAR YMAN{IPULATOR} BY n

RUN

Tells the program to start taking data in the mode pre-determined by the parameters on the status table.

SET

Allows the operator to set various parameters

SET HELP

Lists arguments to the SET command.

SET SCALE HISTnn

Sets various parameters of the display of histogram nn, where nn is a number from 01 to 99.

SET SCALE HISTnn XLO -2.75

Sets the lower limit of the x-axis on the display of histogram nn to the floating point number indicated, -2.75 in the example.

SET SCALE HISTnn XHI 52.7

Sets the upper limit of the x-axis.

SET SCALE HISTnn YLO 0.0

Sets the lower limit of the y-axis.

SET SCALE HISTnn YHI 32.0

Sets the upper limit of the y-axis.

SET SCALE HISTnn LOG

Makes the display of histogram nn a semi log plot.

SET SCALE HISTnn LIN

Returns the display of histogram nn to the default linear plot.

SET SCALE HISTnn AUTOX

Tells the program to select x scales which include all the data. (default)

SET SCALE HISTnn AUTOY

Tells the program to select y scales which include all the data. (default)

SET SCALE GRPHnn

Serves the same purpose as SET SCALE HISTnn except it changes parameters of graph nn where nn is a number from 01 to 99. The syntax is identical.

SET SCALE GRPHnn XLO 23.

SET SCALE GRPHnn XHI 57.

SET SCALE GRPHnn YLO 2.E-3

SET SCALE GRPHnn YHI 1.E5

SET SCALE GRPHnn LOG

SET SCALE GRPHnn LIN

SET SCALE GRPHnn AUTOX

SET SCALE GRPHnn AUTOY

SET ALL MANIP{ULATORS} TO Ø

Zeros the readings for the positions of the X and Y manipulators. This command is used to set the origin of the XY grid, equivalent to:

SET NEAR XMAN TO Ø

SET FAR XMAN TO Ø

SET NEAR YMAN TO Ø

SET FAR YMAN TO Ø

SET STATUS filnam.ext

Tells the program to read in a disk file by the name filnam.ext to place its contents into the status tables. The user may find it convenient to set up such tables ahead of time for certain standard configurations.

SET STATUS n TO idat

Sets the status word, n, to the integer, idat. Use the command SHOW STATUS to obtain the meaning of each status word.

SHOW

Displays a specified table on the terminal.

SHOW HELP

Lists the arguments to the SHOW command.

SHOW SCALE

Lists all scale parameters for the histograms and graphs on the terminal. (See the SET SCALE HISTnn and SET SCALE GRPHnn command for more information.)

SHOW POSI{TION}

Displays the encoder count position of the cart and the X and Y manipulators. The current status of the limit switches is also shown.

SHOW DATA

Lists the table of data stored in the computer memory.

SHOW STATUS

Lists the parameters currently set on the status table.

STOP

Tells the computer to abort the RUN command. The STOP command may be used if the user realizes he has set up the status table incorrectly and doesn't want to wait while the apparatus acquires a lot of irrelevant data. Otherwise, the run will end automatically when the measurement menu set up in the status table is completed.

TAPE

Allows the user to manipulate and set up the status of the magnetic tape.

TAPE RWND

Rewinds tape back to the first record.

TAPE READ n

Reads n records from the tape where n is a positive integer.

TAPE BKSP n

Positions the tape backwards n records, where n is a positive integer. This command can be used to read a particular record.

TIME

Displays at the terminal, the current time. (As accurate as initial value given in booting sequence.)

OVERLAY STRUCTURE

ROOT SEGMENT

File Name	Contents
ZPMAIN	MAIN,EXITR,NF2ND,HZERO
COMMND	COMMND
UPLOT	
TTYIO	
CLOCK	

Internal stuff for data acquisition for anything that must read transient data.

SEG. 1

HISTOL	HISTOL

Data acquisition stuff not in root such as

ZPANAL	ANAL

SEG. 2

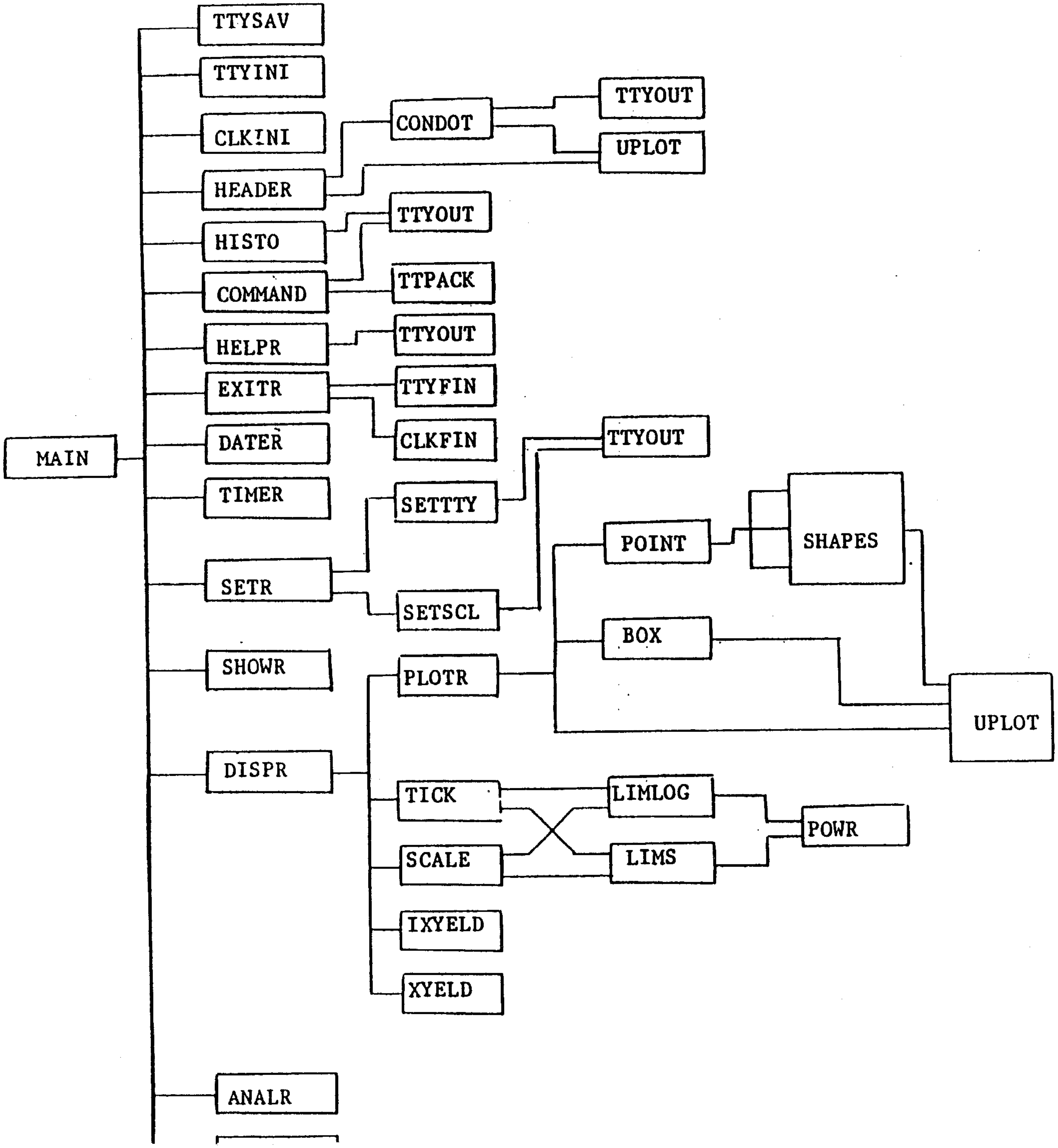
HISTO	HISTO
ZPUTIL	SHOWR,TIMER,DATER,SETR,SETTTY, SETSCL,HELPR,SETSTA

SEG. 3

HISTOB	HISTOB

SEG. 4

DISP	HEADER,CONDOT,DISPR,PLOTR,TICK,IXYELD,SCALF,XYELD, LIMLOG,LIMS,POWR,BOX,POINT,shapes


MAIN
TTYSAV
TTYINI
CLKINI
HEADER
HISTO
COMMAND
HELPR
EXITR
DATER
TIMER
SETR
SHOWR
DISPR
ANALR
CONDOT
TTYOUT
UPLOT
TTYOUT
TTPACK
TTYOUT
TTYFIN
CLKFIN
SETTTY
SETSCL
TTYOUT
POINT
SHAPES
BOX
PLOTR
UPLOT
TICK
SCALE
LIMLOG
LIMS
POWR
IXYELD
XYELD

Directions to put Ziptrk on line

	1						1			0			0			0	OCTAL number
SET COMPUTER TO	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BIT number
					o	o	—	o	o	o	o	o	o	o	o	o	0 = lever dow
																	= lever up
levers	up				down		up	down			down						

TURN ON POWER AT TOP OF RACK (PRESS IN)

TURN KEY TO POWER

	ENABLE,	LOAD ADRS,	START
BOOT	down	down	press down

?KMON-F-COMMAND FILE NOT FOUND

type DATE - date-month-year (for example: 19-JAN-83)

TIME - hour:minutes (for example: 13:51)

type RUN ZIPTRK
(floppy disk unit 0)

ON-LINE EXAMPLES

User-entered commands are shown underlined and in capital letters, computer response in capital letters.

Help Commands

HELP (cr)

COMMANDS ARE:

HELP
SET
EXIT
DATE
TIME
SHOW
DISP
ANAL
RUN
STOP
TAPE
MOVE
CAL

SET HELP (cr)

ARGUMENTS TO SET ARE:

HELP
TTY
SCAL
STAT

SHOW HELP (cr)

ARGUMENTS TO SHOW ARE:

SCALE, DATA, STATUS, OR POSITION

DISP HELP (cr)

ARGUMENTS TO DISPLAY ARE:

HISTnn, GRPHnn, OR OEDn

type - SHOW POSI

	POSITIONS	
DEVICE	ENCODER READING	LIMIT SWITCH
CART	-7	0
NEAR X MANIP	0	0
NEAR Y MANIP	0	0
FAR X MANIP	0	0
FAR Y MANIP	0	0

To set manipulators to zero

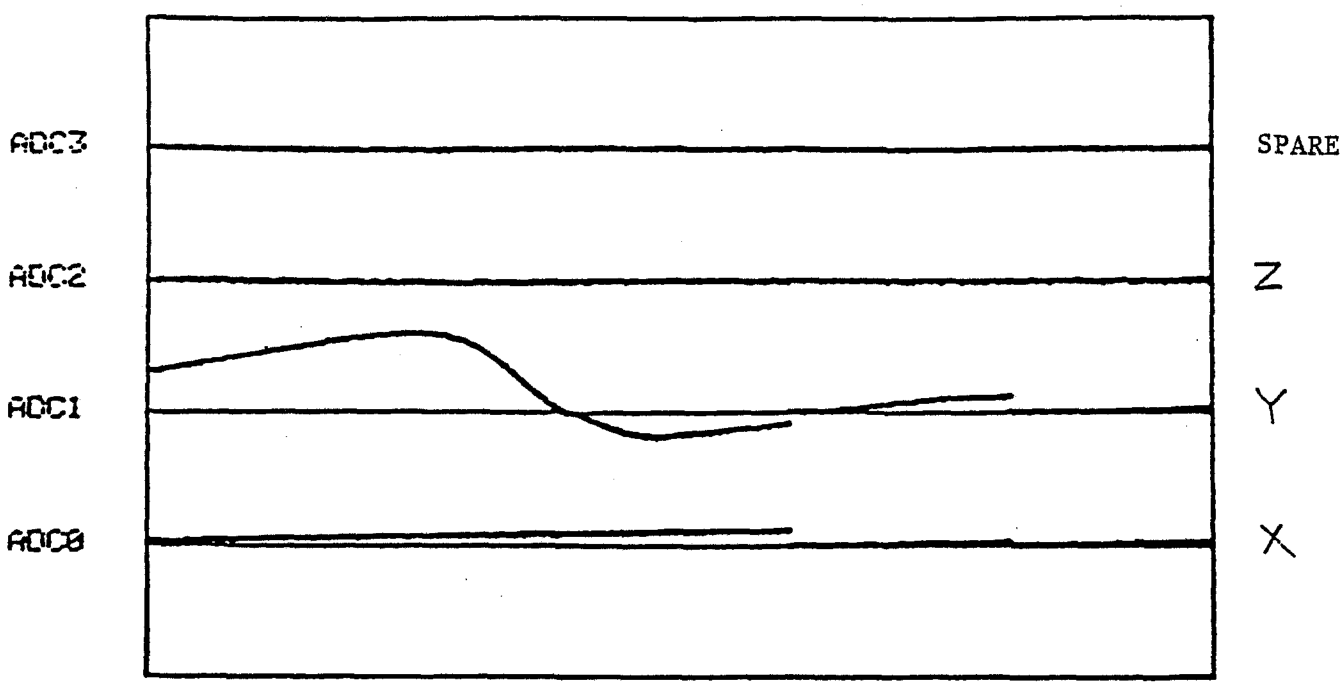
Type - SET ALL MANI TO Ø

or - SET XMANI TO Ø

etc

type - <u>CAL</u>

To calibrate integrators, press the RESET button on the integrators and turn the ZERO ADJUST knob to lower or raise the beam to its corresponding line on the chart (i.e., "ADCØ" for adjusting the X integrator). Press reset and give the CAL command as many times as is necessary to keep all integrators along the respective lines. Turn drift zero knob clockwise to lower and counterclockwise to raise.



type - SHOW STAT

STATUS TABLE

WORD	CONTENTS	MEANING	
1	1	RUN NUMBER	for your particular run
2	1	MAG TAPE FILE NUMBER	
3	3000	MAGNET CURRENT	
4	1	CURRENT DIRECTION (+1 OR -1)	
5	5	X GRID POINTS TOTAL	
6	7	Y GRID POINTS TOTAL	
7	100	NEAR X ENCODER COUNTS PER GRID POINT	
8	100	NEAR Y ENCODER COUNTS PER GRID POINT	
9	100	FAR X ENCODER COUNTS PER GRID POINT	
10	100	FAR Y ENCODER COUNTS PER GRID POINT	
11	50	NEAR X ENCODER COUNT FOR LEFT-MOST PNT	
12	60	NEAR Y ENCODER COUNT FOR LOWEST PNT	
13	50	FAR X ENCODER COUNT FOR LEFT-MOST PNT	
14	60	FAR Y ENCODER COUNT FOR LOWEST PNT	
15	1	RUN MODE (1-2)	1 is manual, 2 is auto
16	100	CART ENCODER COUNT INCREMENT FOR DATA	
17	3	NO. OF OPTICAL SWITCHES (CHANNEL 1+2)	
18	0	FLAG TO LOG DATA TO TAPE (0=NO, -1=YES)	
19	1	ADC SCALE (1,2,4 OR 8)	
20	500	NEARX STEPPING MOTOR PULSES PER 360 ROTA	
21	400	NEARY STEPPING MOTOR PULSES PER 360 ROTA	
22	400	FAR X STEPPING MOTOR PULSES PER 360 ROTA	
23	400	FAR Y STEPPING MOTOR PULSES PER 360 ROTA	
24	1000	NEARX ENCODER COUNTS PER 360 ROTATION	
25	1000	NEARY ENCODER COUNTS PER 360 ROTATION	
26	1000	FAR X ENCODER COUNTS PER 360 ROTATION	
27	1000	FAR Y ENCODER COUNTS PER 360 ROTATION	

to change - STATUS TABLE

type - SET STAT ____ TO ________
word contents

STATUS TABLE

WORD	CONTENTS	MEANING
1	1	RUN NUMBER
2	1	MAG TAPE RECORD NUMBER
3	3000	MAGNET CURRENT
4	1	CURRENT DIRECTION (+1 OR -1)
5	5	X GRID POINTS TOTAL
6	7	Y GRID POINTS TOTAL
7	100	NEAR X ENCODER COUNTS PER GRID POINT
8	100	NEAR Y ENCODER COUNTS PER GRID POINT
9	100	FAR X ENCODER COUNTS PER GRID POINT
10	100	FAR Y ENCODER COUNTS PER GRID POINT
11	50	NEAR X ENCODER COUNT FOR LEFT-MOST PNT
12	60	NEAR Y ENCODER COUNT FOR LOWEST PNT
13	50	FAR X ENCODER COUNT FOR LEFT-MOST PNT
14	60	FAR Y ENCODER COUNT FOR LOWEST PNT
15	1	RUN MODE (1-2)
16	100	CART ENCODER COUNT INCREMENT FOR DATA
17	3	NO. OF OPTICAL SWITCHES (CHANNEL 1+2)
18	0	FLAG TO LOG DATA TO TAPE (0=NO, -1=YES)
19	1	ADC SCALE (1,2,4 OR 8)
20	500	NEARX STEPPING MOTOR PULSES PER 360 ROTA
21	400	NEARY STEPPING MOTOR PULSES PER 360 ROTA
22	400	FAR,X STEPPING MOTOR PULSES PER 360 ROTA
23	400	FAR Y STEPPING MOTOR PULSES PER 360 ROTA
24	1000	NEARX ENCODER COUNTS PER 360 ROTATION
25	1000	NEARY ENCODER COUNTS PER 360 ROTATION
26	1000	FAR X ENCODER COUNTS PER 360 ROTATION
27	1000	FAR Y ENCODER COUNTS PER 360 ROTATION

t SET STAT 5 TO 2 — 2 or number desired } for grid
y SET STAT 6 TO 2 — 2 or number desired } for grid
p SET STAT 15 TO 2 — for auto
e SET STAT 17 TO 5 — depends on length of beam

type - SHOW STAT - shows new stat

STATUS TABLE

WORD	CONTENTS	MEANING
1	1	RUN NUMBER
2	1	MAG TAPE RECORD NUMBER
3	3000	MAGNET CURRENT
4	1	CURRENT DIRECTION (+1 OR -1)
5	2	X GRID POINTS TOTAL
6	2	Y GRID POINTS TOTAL
7	100	NEAR X ENCODER COUNTS PER GRID POINT
8	100	NEAR Y ENCODER COUNTS PER GRID POINT
9	100	FAR X ENCODER COUNTS PER GRID POINT
10	100	FAR Y ENCODER COUNTS PER GRID POINT
11	50	NEAR X ENCODER COUNT FOR LEFT-MOST PNT
12	60	NEAR Y ENCODER COUNT FOR LOWEST PNT
13	50	FAR X ENCODER COUNT FOR LEFT-MOST PNT
14	60	FAR Y ENCODER COUNT FOR LOWEST PNT
15	1	RUN MODE (1-2)
16	100	CART ENCODER COUNT INCREMENT FOR DATA
17	2	NO. OF OPTICAL SWITCHES (CHANNEL 1+2)
18	0	FLAG TO LOG DATA TO TAPE (0=NO, -1=YES)
19	1	ADC SCALE (1,2,4 OR 8)
20	500	NEARX STEPPING MOTOR PULSES PER 360 ROTA
21	400	NEARY STEPPING MOTOR PULSES PER 360 ROTA
22	400	FAR X STEPPING MOTOR PULSES PER 360 ROTA
23	400	FAR Y STEPPING MOTOR PULSES PER 360 ROTA
24	1000	NEARX ENCODER COUNTS PER 360 ROTATION
25	1000	NEARY ENCODER COUNTS PER 360 ROTATION
26	1000	FAR X ENCODER COUNTS PER 360 ROTATION
27	1000	FAR Y ENCODER COUNTS PER 360 ROTATION

Typing RUN now would only send the cart down the track once, since run mode (word 15) is set at 1 (manual).

If run mode were set at 2, the current status table would send the cart on 4 runs, upon entering RUN. Words 7-10 determine the distances between the runs while words 11-14 set the position of the first run.

If word 18 were at -1 the data recorded on the runs would be written on tape.

Distance (in the z direction) between the reading is determined by word 16.

As an alternate:

To change STAT for your test use TECO as follows:

Boot

o RUN TECO

* ERSTATUS.ZIP $ $	$ is altmode (or escape key)
* EWSTATUS.nnn $ $	nnn is your experiment number
* P $ $	(reads STATUS. ZIP program)

Now use some of the following commands to establish your STAT:

* 10V $ $	(displays 1st 10 lines)
* 10L $ $	(moves pointer down 10 lines)
* 1L $ 4V $ $	(moves pointer down 1 line views 4 lin centered on pointer)
* nT $ $	(prints a line from the pointer to the end of the nth line)
* T $ $	(prints one line from pointer)
* FS _ _ _existing_ $_ _change line $ 4V $ $	(searches for specified line and chan it and prints 4 lines centered or change line)
* S _ _ _ _ _ _ _$ 4V $ $	(search for _ _ _ _)
* T _ _ _ _ _ _ _$ $	(print _ _ _ _ _ _ _)
* ØL $ FS _ _ _ _$ _ _ _ $ 4V $ $	(moves pointer to beginning of line, otherwise same as above)
* B $ $	(goes to top of page)

To conclude change of STAT, type:

* EX $ $	(ends program)
* CNTL C	(kills program)

In the ZIPTRK program use the following to use your STATUSnnn:

R ZIPTRK

SET STAT STATUS.nnn (instead of SHOW STAT)

TO READ TAPE

A. To read the tape (n records starting from the beginning)

TAPE RWND
TAPE READ n (where n is a positive integer)

B. To read a particular record on the tape (for example to read record 3 after 5 taping records) - - - - -

TAPE BKSP 3
TAPE READ 1

then to resume taping

TAPE READ 2

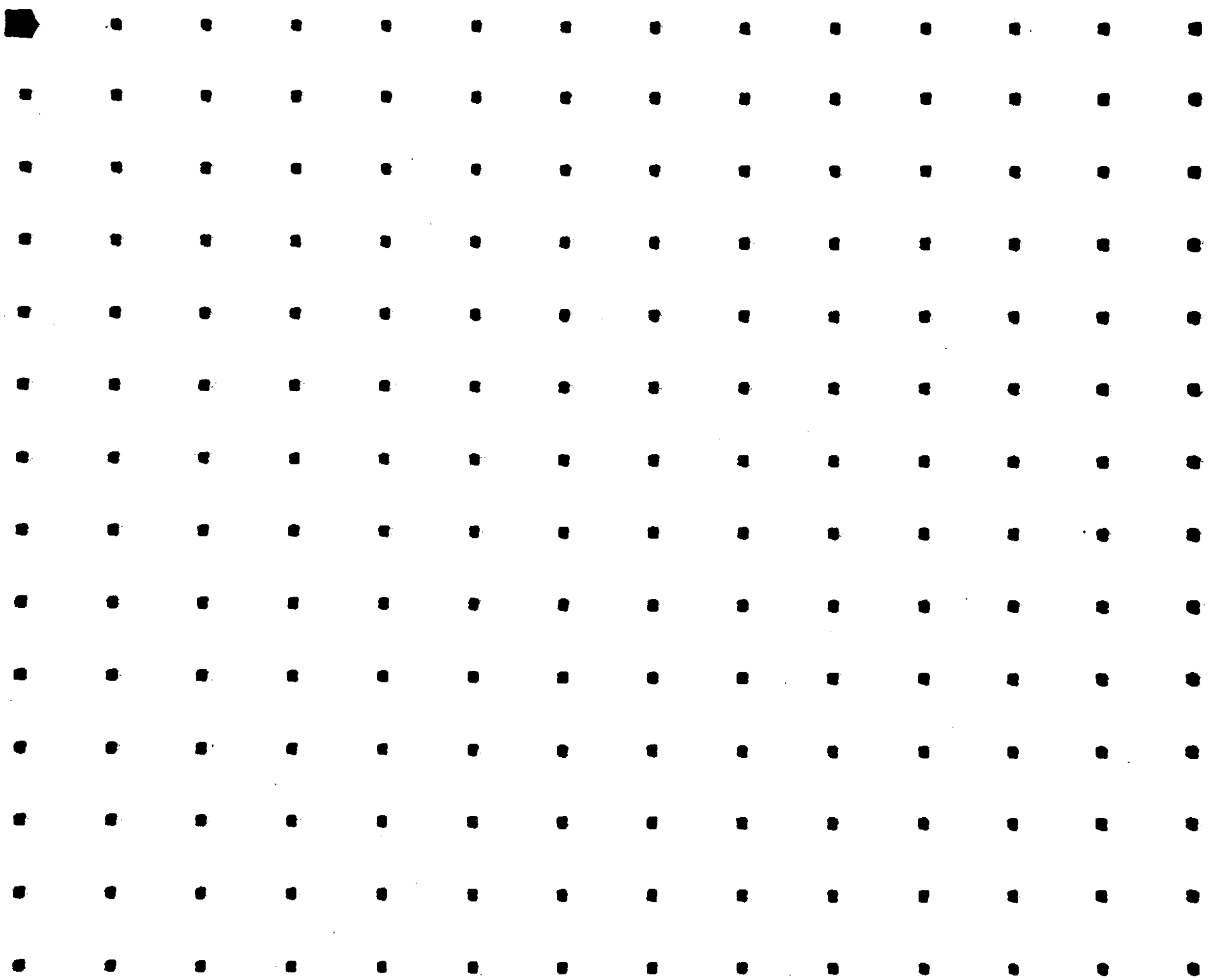
C. The data from the last record read is stored in the computer memory.

To display the data type

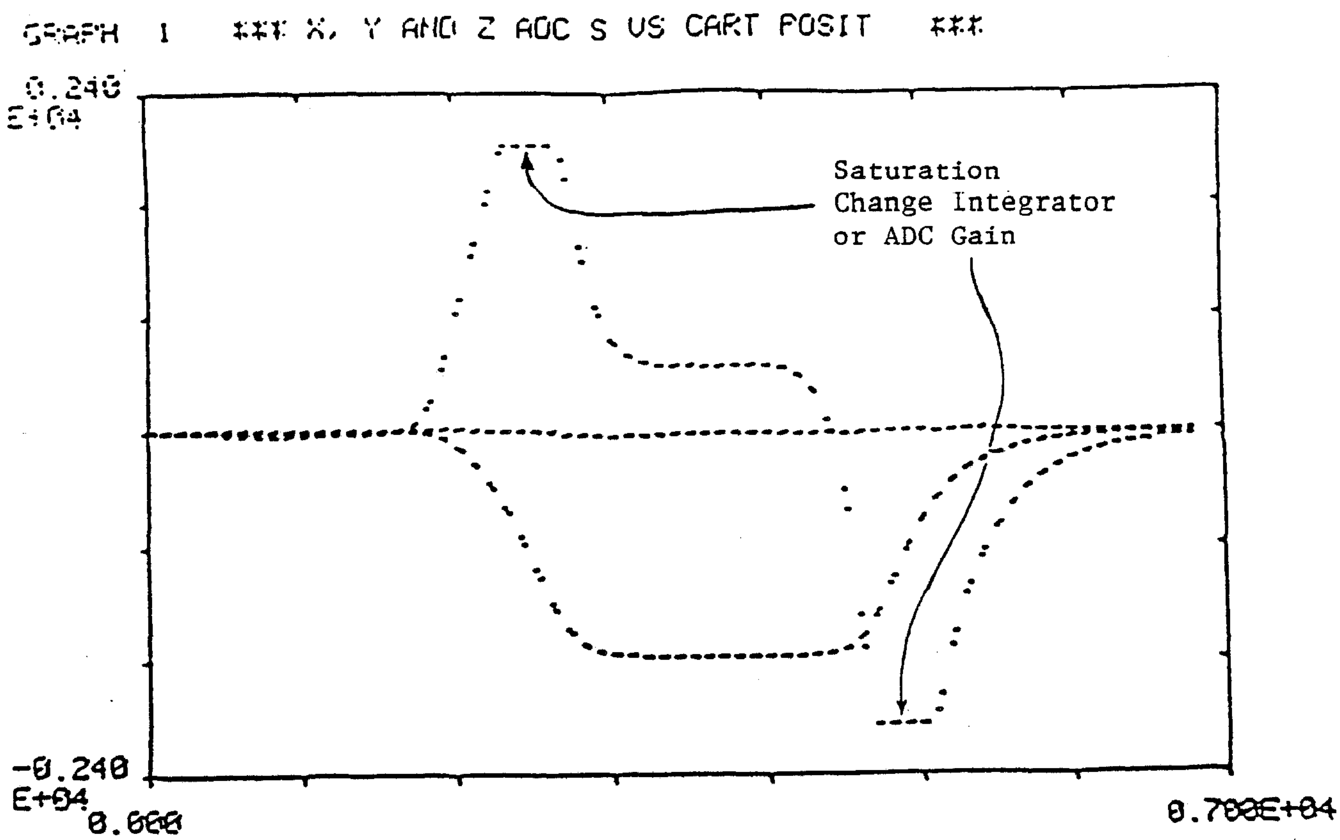
SHOW DATA

type - DISP OED1

Shows current position of beam in a grid of all positions to be run.



type - DISPL GRAPHØ1



type	to get
DISP GRPHØ1	X, Y and Z. ADC's VS CART POSITION
DISP GRPHØ2	X ADC VS CART POSITION
DISP GRPHØ3	Y ADC VS CART POSITION
DISP GRPHØ4	Z ADC VS CART POSITION

type - SHOW DATA

```
STATUS TABLE
  27  NO. OF WORDS IN STATUS TABLE
     1  RUN NUMBER
     1  MAG TAPE RECORD NUMBER
  3000  MAGNET CURRENT
     1  CURRENT DIRECTION (+1 OR -1)
     2  X GRID POINTS TOTAL
     2  Y GRID POINTS TOTAL
   100  NEAR X ENCODER COUNTS PER GRID POINT
   100  NEAR Y ENCODER COUNTS PER GRID POINT
   100  FAR  X ENCODER COUNTS PER GRID POINT
   100  FAR  Y ENCODER COUNTS PER GRID POINT
    50  NEAR X ENCODER COUNT FOR LEFT-MOST PNT
    60  NEAR Y ENCODER COUNT FOR LOWEST PNT
    50  FAR X ENCODER COUNT FOR LEFT-MOST PNT
    60  FAR Y ENCODER COUNT FOR LOWEST PNT
     1  RUN MODE (1-2)
   100  CART ENCODER COUNT INCREMENT FOR DATA
     2  NO. OF OPTICAL SWITCHES (CHANNEL 1+2)
     0  FLAG TO LOG DATA TO TAPE (0=NO, -1=YES)
     1  ADC SCALE (1,2,4 OR 8)
   500  NEARX STEPPING MOTOR PULSES PER 360 ROTA
   400  NEARY STEPPING MOTOR PULSES PER 360 ROTA
   400  FAR X STEPPING MOTOR PULSES PER 360 ROTA
   400  FAR Y STEPPING MOTOR PULSES PER 360 ROTA
  1000  NEARX ENCODER COUNTS PER 360 ROTATION
  1000  NEARY ENCODER COUNTS PER 360 ROTATION
  1000  FAR X ENCODER COUNTS PER 360 ROTATION
  1000  FAR Y ENCODER COUNTS PER 360 ROTATION

MANIPULATOR ENCODERS NEAR X, NEAR Y, FAR X, FAR Y
        0       0       0       0

     0  NMR ADC READING
```

(continued on next page)

OPTICAL SWITCH ENCODER READINGS
2 NO. OF OPTICAL SWITCH READINGS
4508 ENCODER COUNTS AT OPTO SWITCH CROSSING # 1
4519 ENCODER COUNTS AT OPTO SWITCH CROSSING # 2
2 ENCODER COUNTS AT OPTO SWITCH CROSSING # 3

ADC DATA
90 NO. OF READINGS DOWN AND BACK

DATA PNT	ENCODER	ADC0	ADC1	ADC2
1	100	19	4	-102
2	200	24	3	-133
3	300	28	3	-161
4	400	32	3	-190
5	500	36	3	-218
6	600	40	3	-247
7	700	44	3	-272
8	800	48	5	-302
9	900	52	6	-331
10	1000	56	6	-360
11	1100	60	7	-389
12	1200	63	8	-419
13	1300	68	9	-448
14	1400	72	9	-477
15	1500	75	11	-507
16	1600	79	11	-533
17	1700	83	12	-562
18	1800	87	12	-592
19	1900	90	13	-622
20	2000	94	14	-652
21	2100	98	15	-681
22	2200	102	16	-709
23	2300	105	17	-738
24	2400	109	18	-768
25	2500	113	18	-794

SYSTEM CALIBRATION

SUMMARY OF ADC CALIBRATION

Polarity	Gain	Counts
Positive	1	204.81*V + .17
	2	409.60*V + .22
	4	819.24*V + .44
	8	1637.76*V + 1.60
Negative	1	204.87*V + .40
	2	409.64*V + .39
	4	819.44*V + .66
	8	1638.54*V + 1.15

V = voltage input to ADC

Counts = counts output from ADC

Coil Calibration
(30' cables)
February 23, 1983

$$\tau_E = \left(\frac{R_E + R_I}{R_I}\right)\tau_I$$

$$R_E = Coil_R + Cable_R$$

X

R_E = 262.3 Ω R_{Coil} = 233.6Ω	τ_I(ms) Intgr T.C.	R_I Intgr Input	τ_E Effective T.C.	K Volts/KG
300 ms	300.62	299.82 KΩ	300.88	0.008023
100	100.70	100.37 KΩ	100.96	0.02391
30	30.16	30.080 KΩ	30.423	0.07935
10*	10.04	10011 Ω	10.3031	0.2343
3	3.033	3023.1	3.2962	0.7324
1	1.007	1003.71	1.2702	1.9006
0.3	0.3009	299.75	.5642	4.2788
0.1	0.1012	100.36	.3657	6.6015

Y

R_E = 141.15Ω R_{coil} = 114.7Ω				
300 ms	301.38	300.24 KΩ	301.5217	0.02863
100	100.97	100.57 KΩ	101.1117	0.08538
30*	30.19	30.079 KΩ	30.3317	0.2846
10	10.04	10001.7 Ω	10.1817	0.8479
3	3.025	3012.4	3.1667	2.7262
1	1.010	1005.85	1.1517	7.4959
0.3	0.3020	299.82	0.4442	19.437
0.1	0.1010	100.24	0.2432	35.852

(30' cables)

$R_E = 27.56\,\Omega$

Z $R_{coil} = 3.19\,\Omega$

300 ms	311.33	300.42 K	311.359	0.03530
100	104.11	100.423 K	104.139	0.1056
30*	31.19	30.095 K	31.219	0.3521
10	10.39	10016.4	10.419	1.0550
3	3.143	3031.3	3.1716	3.4656
1	1.039	1001.53	1.0676	10.296
0.3	0.3118	300.37	0.3404	32.289
0.1	0.1046	100.37	0.13332	82.450

Coil Calibration
(100' cables)
June 24, 1983

$$\tau_E = \left(\frac{R_E + R_I}{R_I}\right) \tau_I$$

$$R_E = Coil_R + Cable_R$$

X

$R_E = 263\ \Omega$ $R_{coil} = 233.6\ \Omega$	τ_I (ms) Intgr T.C.	R_I Intgr Input	τ_E (ms) Effective T.C.	K Volts/KG
300 ms	300.62	299.82 KΩ	300.88	0.008020
100	100.70	100.37 KΩ	100.96	0.02390
30	30.16	30.080 KΩ	30.424	0.07931
10	10.04	10011 Ω	10.3038	0.2342
3	3.033	3023.1	3.2969	0.7319
1	1.007	1003.71	1.2709	1.8987
0.3	0.3009	299.75	.5649	4.2714
0.1	0.1012	100.36	.3664	6.5866

Y

$R_E = 142\ \Omega$ $R_{coil} = 114.7\ \Omega$				
300 ms	301.38	300.24 KΩ	301.5225	0.02863
100	100.97	100.57 KΩ	101.1126	0.08537
30	30.19	30.079 KΩ	30.3325	0.2846
10	10.04	10001.7 Ω	10.1825	0.8477
3	3.025	3012.4	3.1676	2.7251
1	1.010	1005.85	1.1526	7.4893
0.3	0.3020	299.82	0.4450	19.396
0.1	0.1010	100.24	0.2441	35.721

(100' cables)

$R_E = 29\ \Omega$

Z $R_{coil} = 3.19\ \Omega$

300 ms	311.33	300.42 K	311.360	0.03536
100	104.11	100.423 K	104.140	0.1057
30	31.19	30.095 K	31.220	0.3526
10	10.39	10016.4	10.420	1.0566
3	3.143	3031.3	3.1731	3.4696
1	1.039	1001.53	1.0691	10.298
0.3	0.3118	300.37	0.3419	32.200
0.1	0.1046	100.37	0.1348	81.659

Fermilab

ADC Calibration

(For Research Services Personnel Only)

1. Pull PDP 11/05 out from rack and remove top cover (4 screws). Make sure that someone guides ribbon cables at the back of the computer, while the rack is being pulled out.

2. Connect the two flat cables, J1 and J2, from ADC board to separate metal box (5" x 8"). Match the arrows to get the proper orientation of the cables.

3. Connect voltage calibrated power supply DVC 8500 to metal box.

4. Turn on PDP 11/05 and boot using load address: 171000_8

5. Computer will type: ?KMON-F-COMMAND
   FILE NOT FOUND

   Respond with: RUN SPØØ26 (cr)
   MODEL DT1712 (cr)

6. Computer types:

Computer types	Respond with
100 KHZ A/D MODULE (Yor N)	Y
IS THE DMA OPTION PRESENT (Y or N)	Y
# OF A/D INPUT CHANNELS (IN OCTAL)	20 (cr)

7. To change vector address (pg 8-3)*

Enter	Computer types	Respond with
000544/	(000300)	320 (cr)

8. Enter: TEST 1 (cr)
   TEST 2 (cr)
   "
   "
   "
   "
   TEST 7 (cr)
   TEST 10 (cr)
   TEST 11 (cr)
   "
   "
   "
   "
   "
   TEST 17 (cr)

9. Type: TEST 22 (cr)

Computer types	Responds with
Enter Channel no.?	any number from 1 to 17 in octal (cr)
Enter Mode bits?	3 (cr) (for gain of 1)
	7 (cr) (for gain of 2)
	10 (cr) (for gain of 4)
	17 (cr) (for gain of 8)

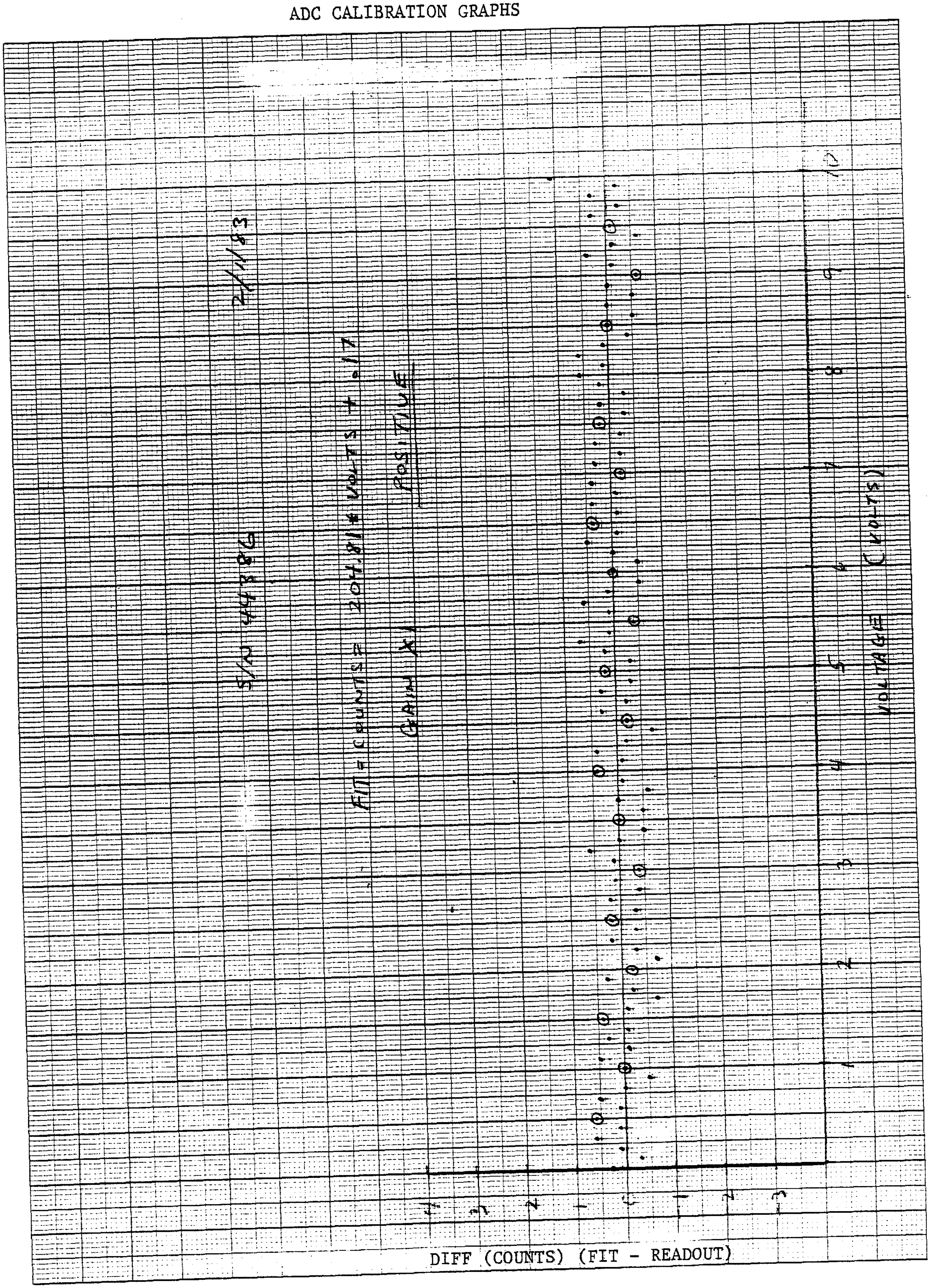
10. Vary the voltage, recording the average number of counts (appearing on screen) for each input voltage. It is not necessary the halt test and retype TEST 22.

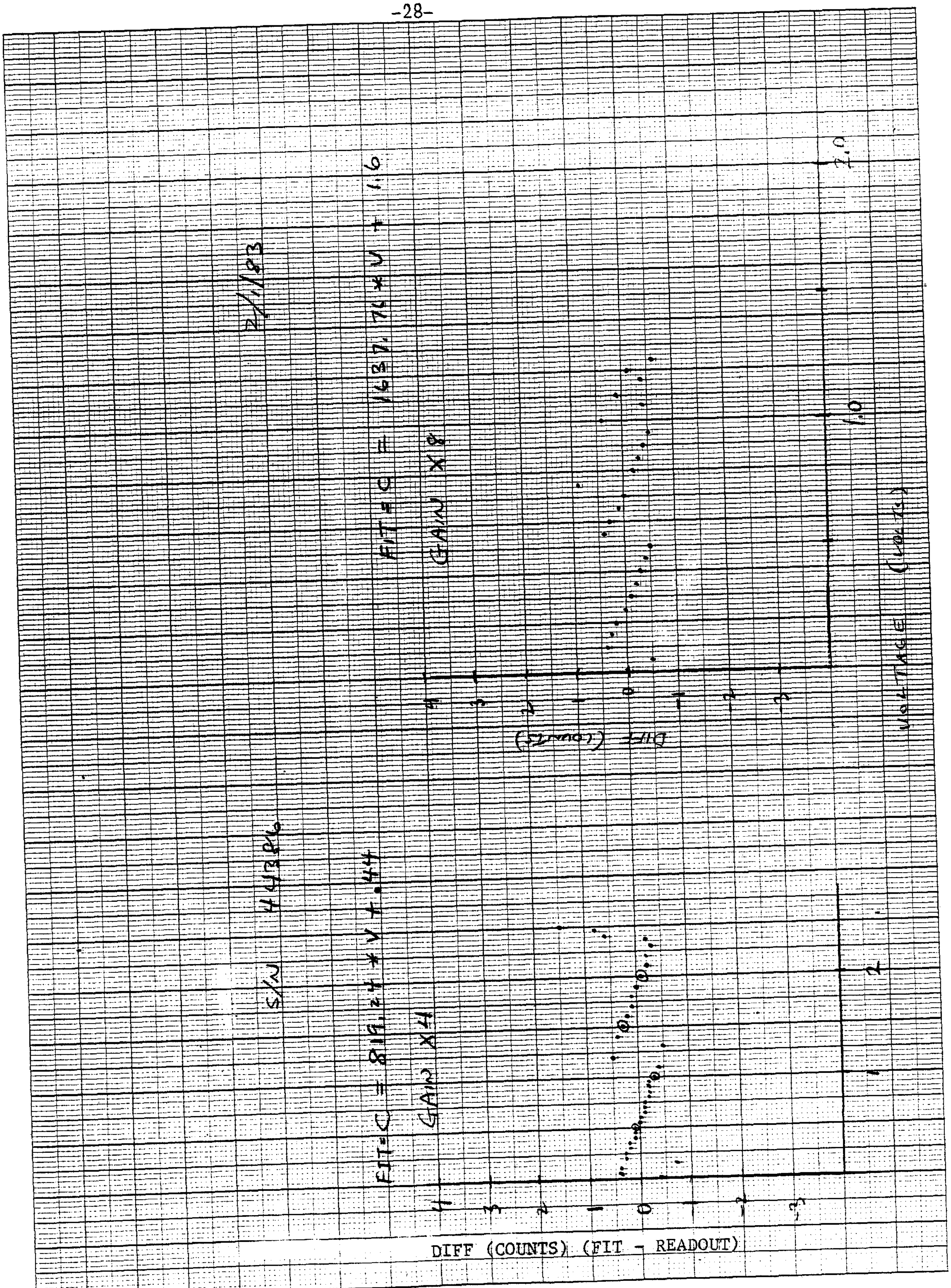
11. Calculate slope of voltage to counts using the least square fit.

12. Plot the result of each gain setting.

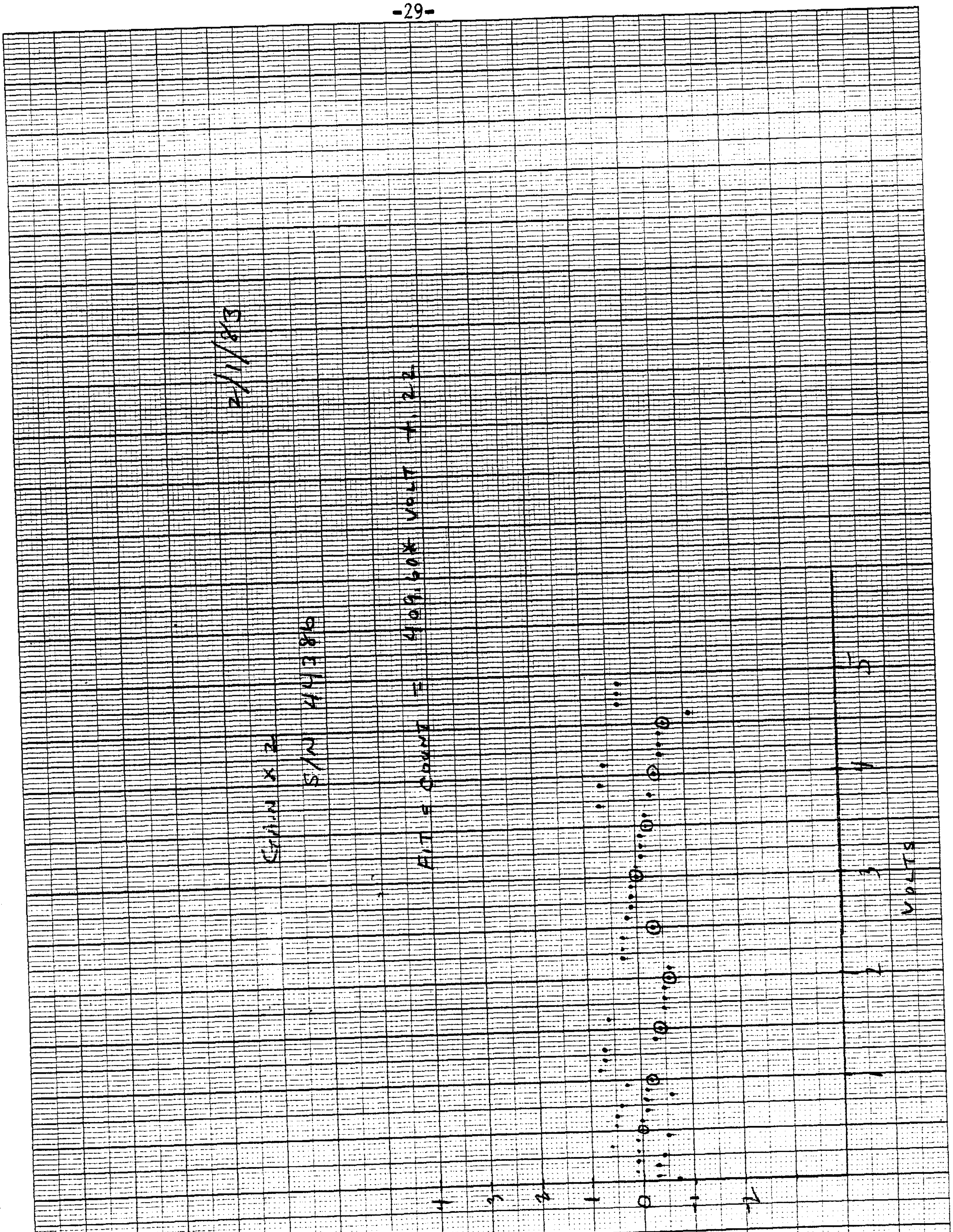
* User Manual for Data Translation, Incorporated, Copyright 1978, Data Translation, 100 Locke Drive, Marlborough, MA 01752

ADC CALIBRATION GRAPHS

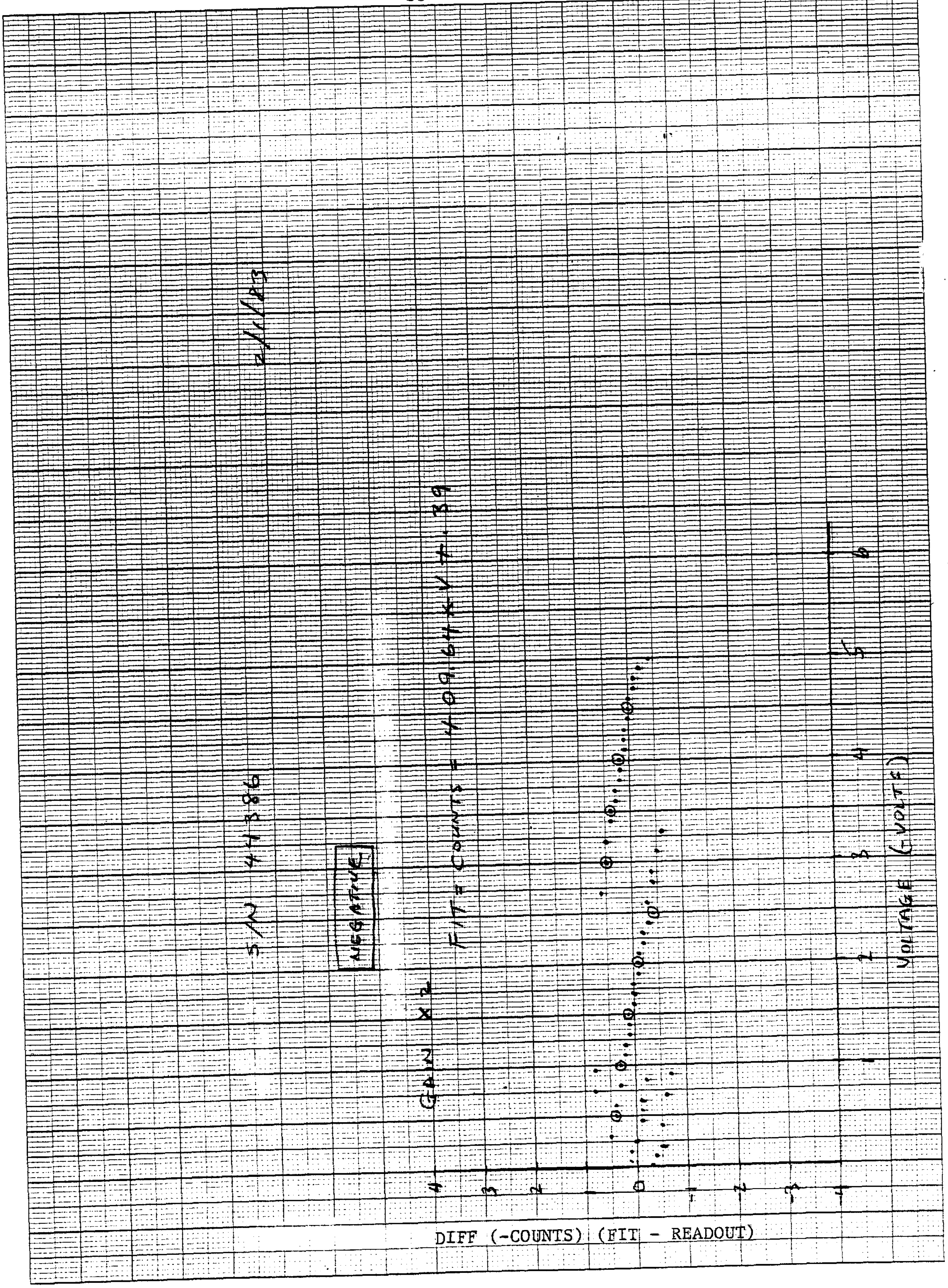


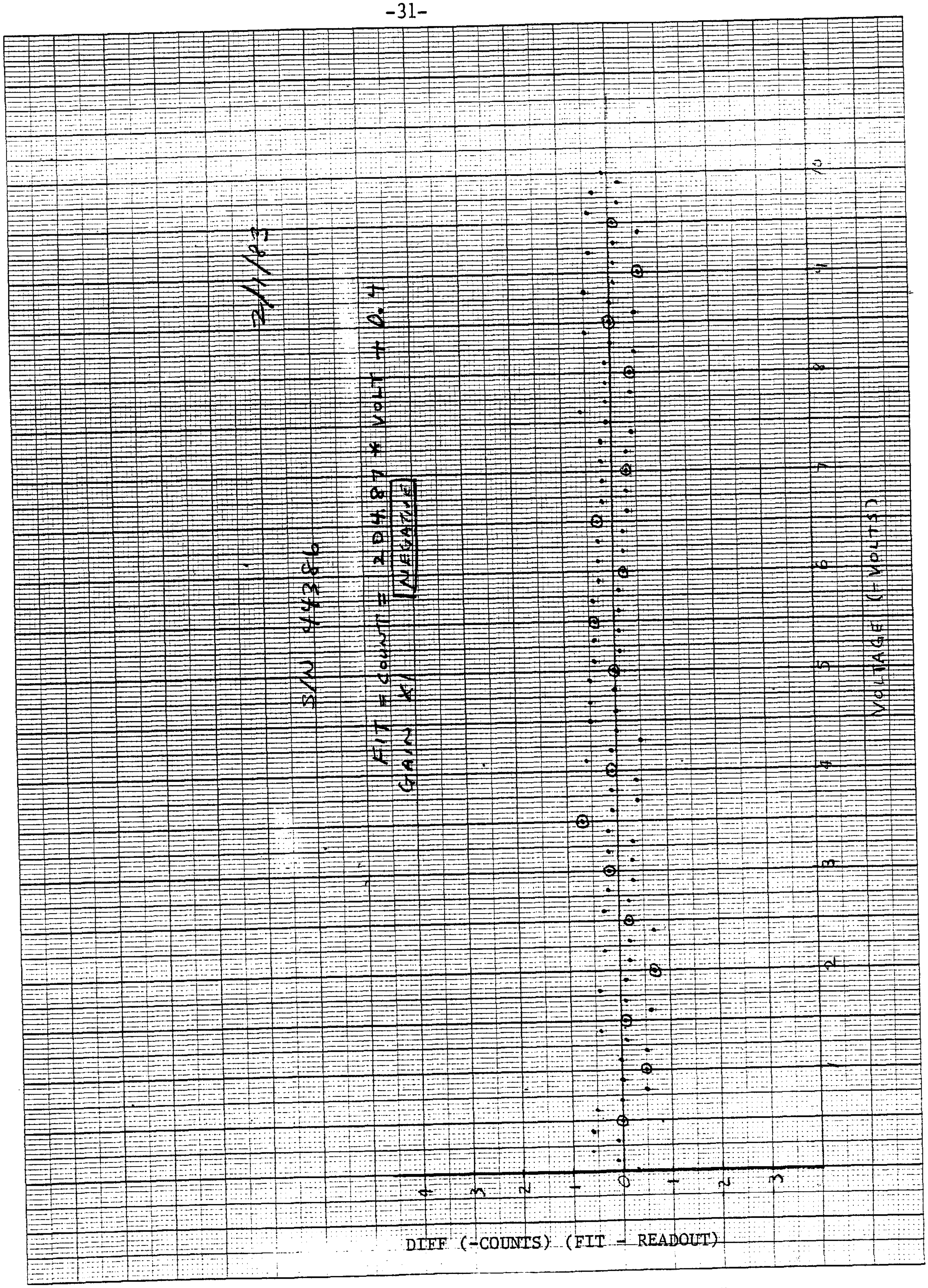


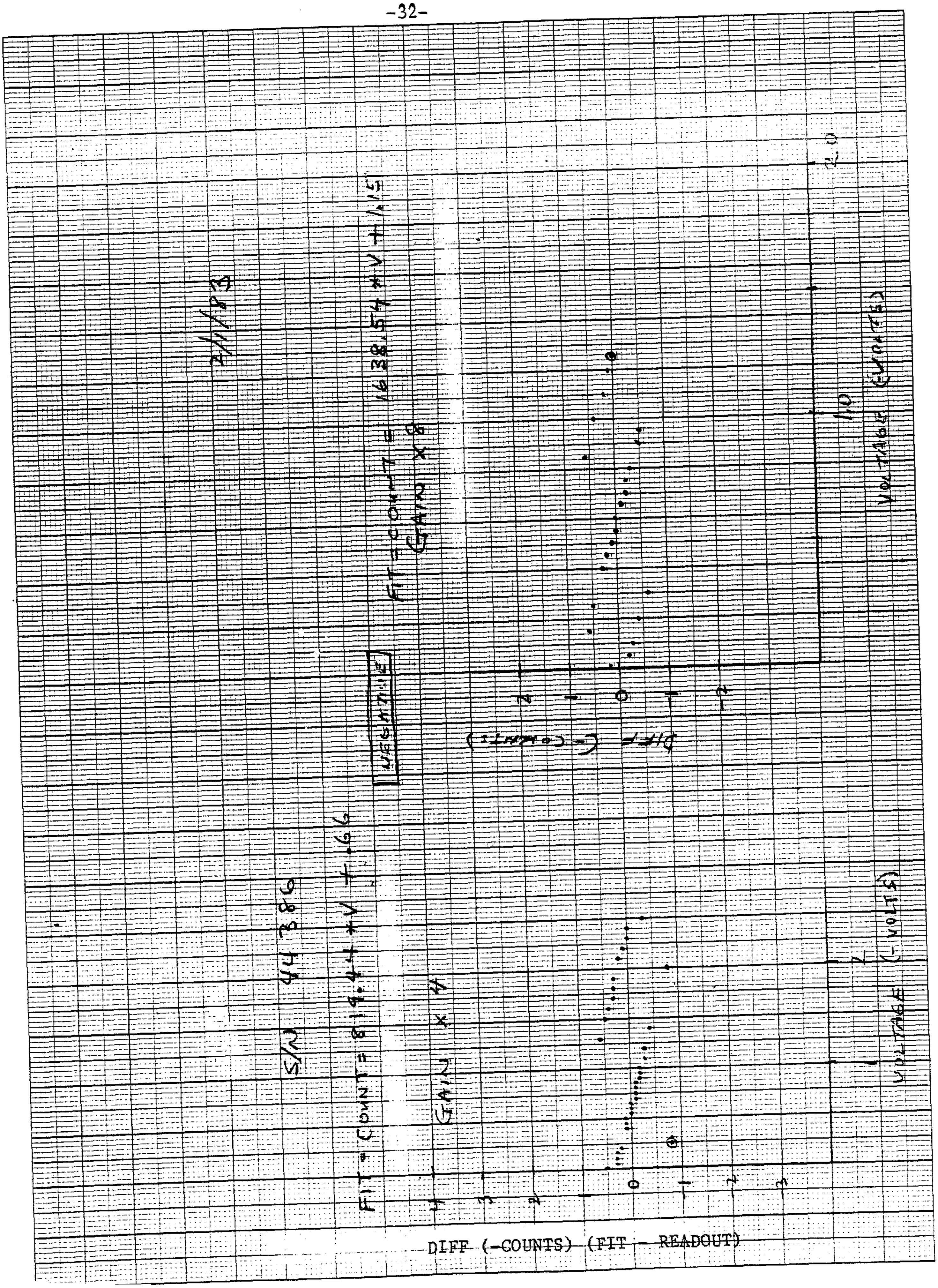
DIFF (COUNTS) (FIT - READOUT)



Cont. DIFF (COUNTS) FIT - READOUT


S/N 44386
2/1/83
NEGATIVE
GAIN X2
FIT = COUNTS = 409.64 K V + .39
VOLTAGE (-VOLTS)
DIFF (-COUNTS) (FIT - READOUT)


S/N 4386
FIT = COUNT = 204.87 * VOLT + 0.4
GAIN X1
NEGATIVE
2/1/83
VOLTAGE (-VOLTS)
DIFF (-COUNTS) (FIT - READOUT)


S/N 44386
2/1/83
FIT = COUNT = 814.44 * V + .66
GAIN X 4
FIT = COUNT = 1638.57 * V + 1.15
GAIN X 8
NEGATIVE
DIFF (-COUNTS)
VOLTAGE (-VOLTS)
1.0
2.0
DIFF (-COUNTS) (FIT - READOUT)

BEAM ALIGNMENT

For elevation and vertical alignment, a WILD N3 and a Brunson are used respectively. Lighted Targets (battery charged) are placed 5" apart on a special drilled face plate on the cart. The cart is moved from one adjusting point to the next so the front wheels align with the adjusting points.

Typically if with the Brunson one obtains the numbers below at an alignment point.

+.013"=> right of line of sight

↑

↓

.026"L =>
left of line of sight

the rail screws at the sides of the adjustment opening are loosened both at the cart wheels and at the adjustment openings forward and behind the cart. The adjusting nuts are then turned as shown with arrows. If at the same time the elevation is low, then there should be more upward adjustment. If the cart is straight, only the elevation is adjusted. The nuts are turned with 5/16" open ended wrenches keeping the rail snug between them. After adjusting, retighten the rail screws behind the cart and move on to the next adjusting position.

Fermilab

Calibration Procedure for Integrators

June 27, 1983

We did the calibration at the Magnet Measurement Facility of Industrial Building 1. We received permission from Bruce Brown for use of the equipment. We brought the integrator and A.C. cord and installed it in the nim bin for a few hours for the integrator to achieve a stable temperature in order to avoid excessive drift. This is very important especially during the cold winter months.

Connect the integrator with the calibration equipment, as shown in Figure 1.

Set Up Procedure

A. Output DVM

1. Press reset

2. To set up 6 digit display:

a. Press 6

b. Press STORE

c. Press "N DIG DISP."

B. Input DVM

1. Press reset

2. To set up 6 digit display:

a. Press 6

b. Press STORE

c. Press "N DIG DISP."

3. To set up number of readings per trigger: (50 readings per trigger in this example)

a. Press 50

b. Press STORE

c. Press "N RD/TRIG"

d. Press "EXT" trigger

e. Press "MATH" (blue)

f. Press "STAT" (blue)

g. Red LED's should be lit beside the STAT, MATH and N RD/TRIG buttons.

C. For the 100 ms scale, scan the integrator with "+" and "-" input voltages by 50 mV step until the output voltage reach $\pm$ 10 V.

D. For the other time constants use the voltages shown in the following chart (postive and negative).

Time Constant (ms)	Input Voltage (+V)	N RD/TRIG	Integration Time (sec)
300	0, 3, 6, 9, 12	50	20
	10	75	30
100	0 to 5.1 V, 50 mV STEP	50	20
30	0, 1, 2, 3, 4	18	7
10	0, 1, 2, 3, 4, 5	5	2
3	0, 0.3, 0.6, 0.9, 1.2, 1.5	5	2
1	0, 0.1, 0.2, 0.3, 0.4, 0.5	5	2
0.3	0, .03, .06, .09, .12, .15, .18, .21	5	2
0.1	0, 20 mV, 40, 60, 80, 100, 120	5	2

Run Procedure

1. Select the integration time
2. Reset the time counter
3. Selection input voltage and polarity
4. Check the DVM display. Note: The voltage divided by 100 is displayed on the DVM.
5. Press "INT" on the output DVM
6. Clamp the integrator and minimize the drift as shown on the DVM. When the drift is at minimum, immediately press "EXT" on output DVM, and press "TRIGGER" on the pulser.
7. Take the Vin and Vout when integration time is over
8. Make sure the readings are in orderly increment
9. Calculate the time constant

$$RC = \frac{(Vin)}{Vout} \quad (\Delta T)$$

For example at 1 V input, 100 ms scale, we have the integration time of 20 seconds.

Vout = 1.97821

Vin = 9.8865×10^{-3}

$$\text{Then RC} = \frac{(9.8865) \times 10^{-3} \times 20}{1.97821} = 99.95 \text{ ms}$$

10. Use least squares fit of the data to calculate the time constants.

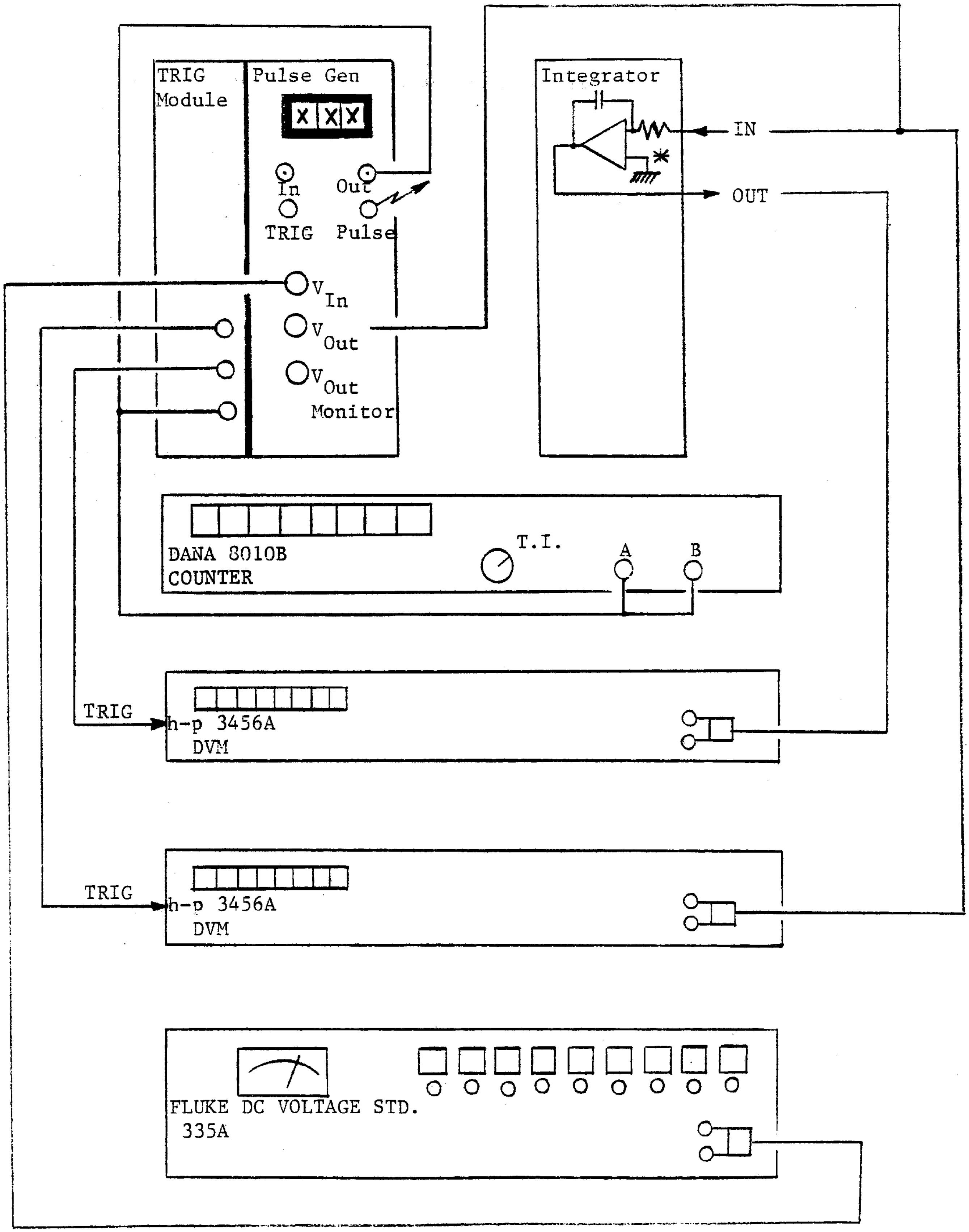
FERMILAB
ENGINEERING NOTE

SECTION	PROJECT	SERIAL-CATEGORY	PAGE
			37

SUBJECT	NAME
INTEGRATOR CALIBRATOR EXTERNAL CONNECTIONS	E. Schmidt

DATE	REVISION DATE
Feb. 10, 1983	

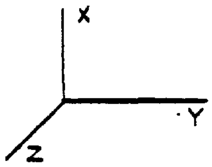
ALL EXTERNAL CONNECTIONS MUST BE FLOATING. CIRCUIT GROUND MADE AT INTEGRATOR *



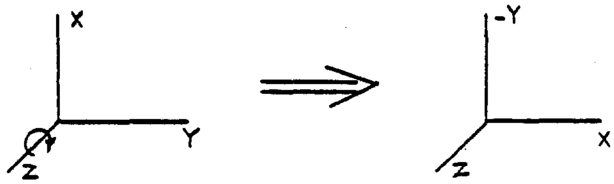
SIGNAL COIL CALIBRATION

Mount the signal coil block on a special jig for rotation. Place the end of the beam (the end farthest from "home") inside the Koehler's Coil Unit. Raise or lower the beam to the appropriate height. Make three runs with the following orientations of the coil:

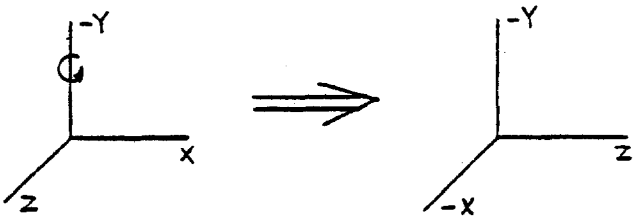
#1 Run as mounted



#2 Rotate coil 90° clockwise about the Z-axis and run.



#3 Rotate coil again, this time by 90° counterclockwise about the y-axis, and run.



<u>Run Procedure</u>

Type: RUN (cr)

Cart should begin moving down the track. When cart is about one foot from the Koehler's Coil Unit, switch on the power supply for the unit. When cart reaches the same spot on its return run, switch off the power. When the run is complete, check the graphs of the run.

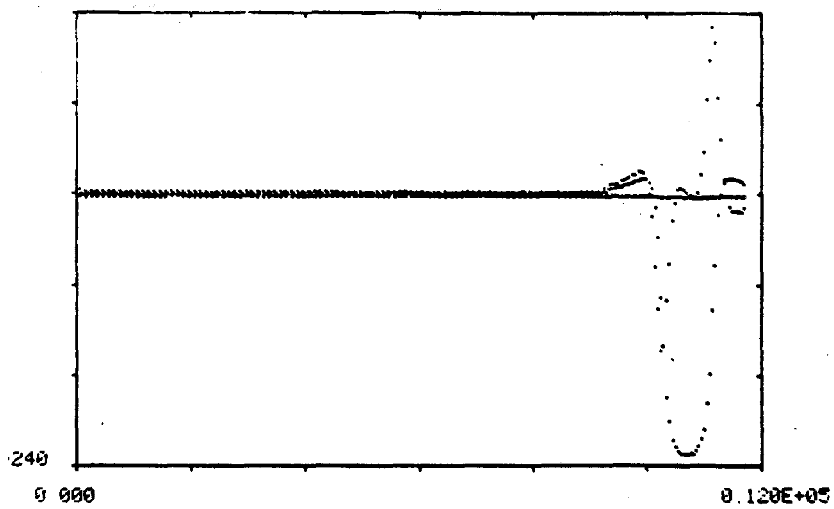
Type: <u>DISP GRPH01</u> (cr)

This should return the upper graph on the attached sheet.

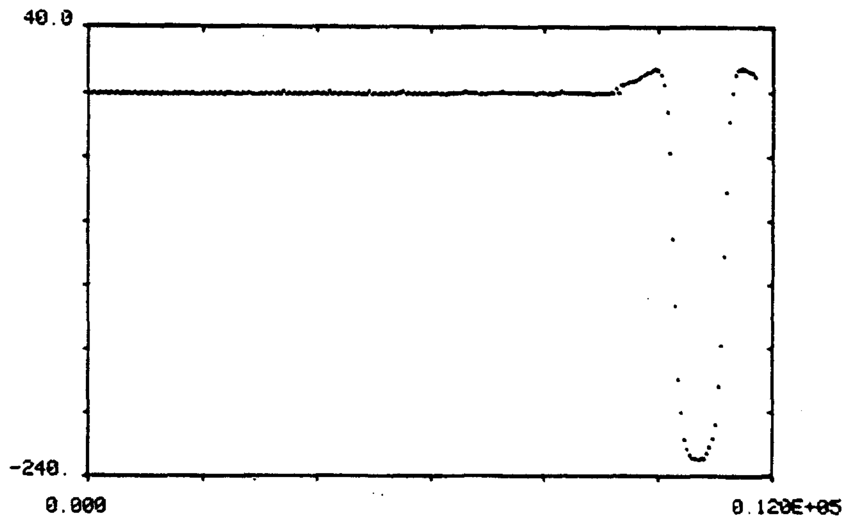
Type: <u>DISP GRPH03</u> (cr)

This should give the lower graph for run #1. (Use GRPH02 for run #2 and GRPH04 for run #3.) If there are two distinct paths traced out in the lower graph, check the calibration of the integrator which corresponds to the coil perpendicular to the field, (y for run #1, x for run #2, and Z for run #3). Also check to see that the signal coil is mounted tightly on the jig. We recommend the following time constants of 0.1, 1.0, and 0.1 ms for x, y, and Z coil, respectively. With ADC gain = 4 for x, and 1 for y and Z.

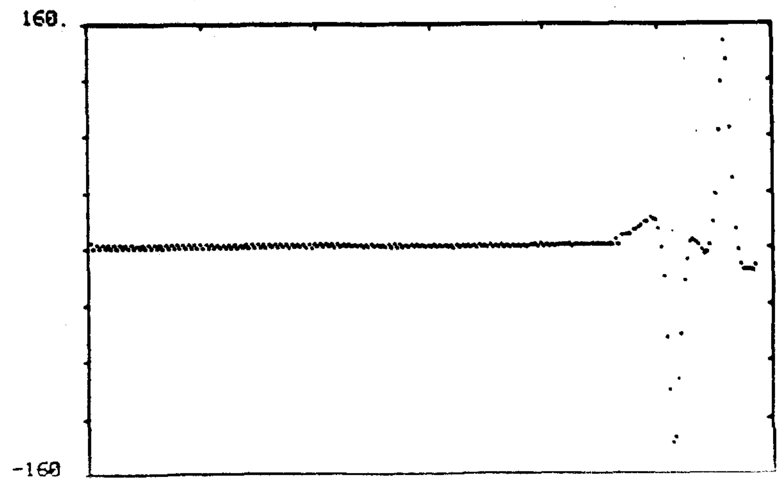
GRAPH 1 *** X Y AND Z ADC S US CART POSIT ***



GRAPH 2 *** X ADC US CART POSITION ***



GRAPH 4 *** Z ADC US CART POSITION ***



Procedure for Computer Control of Manipulator

Make sure the cables are hooked up for the computer to the Buzz Box, for the power supplies, the motors, encoders and limit switches. The starting address is 164100_8. For detail description see A. Lynch's write-up.

(1)

a) Load address 164120_8

b) Hit EXAMINE

Bit 0 to 8 LED should be on: bit 0 to 7 are the limit switches for the horizontal and vertical movements, and bit 8 is for spare. If bit 15 LED is on, it shows the limit switch is hit. It must be cleared. Deposit 100000_8 at this address will clear the bit.

(2)

a) Load address 164102_8

b) Deposit 015530_8 for the rate divisor of channel 1

(3)

a) Load address 164110_8

b) Deposit 015530_8 for the rate divisor of channel 2

(4)

a) Load address 164106_8

b) Deposit 010000_8 for channel 1 motor to move 10,000 encoder counts

(5)

a) Load address 164114_8

b) Deposit 010000_8 for channel 2 motor to move 10,000 encoder counts

(6)

a) Load address 164100_8

b) Deposit 000777 for horizontal movements - watch the movement and the limit switch

or

c) Deposit 001777 for vertical movements

PDP-11 HARDWARE AND INTERFACE

The following three documents describe the functions of the three pdp-11 modules designed specifically for use in the Fermilab Ziptrack magnetic field mapping system. Those three modules are the Cart Position and Control Board, the Manipulator Stepping Motor Controller, and the Manipulator Position Monitor.

Together, the modules position the magnetic field sensor in three dimensions, monitor the sensor's position, and strobe an analogue to digital converter (ADC) when appropriate. In addition, one of the three modules is currently used to zero the integrators for the magnetic field sensor just before the beginning of a data run.

The Cart Position and Control board functions independently of the other two boards. Its function is to cause a motor to move the field sensor cart along its support beam while the sensor's output is being repeatedly digitized by the ADC. This is done while the manipulator remains inactive.

The Cart Position and Control Board also generates the ADC strobe. This signal, which occurs at fixed intervals of cart travel, is the one that triggers the ADC to digitize the output of the magnetic field sensor.

The other two boards function in tandem while the sensor cart remains inactive. The Manipulator Stepping Motor Controller generates a pulse train to the stepping motor drivers. The number and frequency of those pulses determine thedistance and speed of the manipulator move. The particular pulse lines used determine the direction of the motion. Meanwhile, the Manipulator Position Monitor keeps an updated count of the shaft encoder pulses generated by the motion.

The document that follow treat each of the three boards separately and in greater detail.

PROJECT:	GENERAL HARDWARE PROJECTS
DOC. TYPE:	TECHNICAL NOTE
TITLE:	Ziptrack Cart Position and Control Board Documentation
SUBJECT:	
GROUP:	PDP-11 HARDWARE
AUTHOR:	Aaron Lynch
DOC NO.:	40
DATE:	15-JUL-81 15:39:23
FILE:	CART.RNO
BASE ADD:	1642008

1.0 INTRODUCTION

The Ziptrack Cart Position and Control Board is a computer driven D.C. motor controller designed for the positioning of the Ziptrack sensor cart. The board plugs into a standard PDP-11 Unibus backplane required for its proper use.

The board outputs signals causing forward motion, reverse motion, and breaking action of the motor. It receives as input the signals from a shaft encoder connected to the motor, as well as limit switches and optical switches that help to monitor the motor's motion. These input signals are the basis for automatic motor stopping, periodic ADC strobing, and interrupts to the computer.

THE FUNCTION CONTROL REGISTER
ADDRESS OFFSET 16 OCTAL

See figure 1 for a diagrammatic summary of the function control register.

BIT 09: This bit is the master interrupt enable for the controller module. There must be a 1 in this bit in order for any of the module's 4 kinds of interrupts to occur. Also, this bit is automatically cleared during the start of the interrupt itself, so that the software must reload a 1 into this bit in order to enable a subsequent interrupt. All of the interrupts of the controller module carry the same interrupt vector.

THE PERIODIC INTERRUPT FUNCTION

BIT 00: A 1 in this bit along with a 1 in bit 09 enables the periodic interrupt function of the controller. Address offset 14 octal must also be loaded with the number N described below.

The periodic interrupt function consists of a controller gernerated interrupt occurring every N shaft encoder net counts. That is, the controller board automatically increments the divide by N counter every time it receives a clockwise pulse from the shaft encoder, and it decrements the same counter each time it receives a counter-clockwise pulse from the shaft encoder. Then, when this 16-bit counter either borrows or carries, it generates an interrupt and reloads itself with the number N which was stored in the divide by

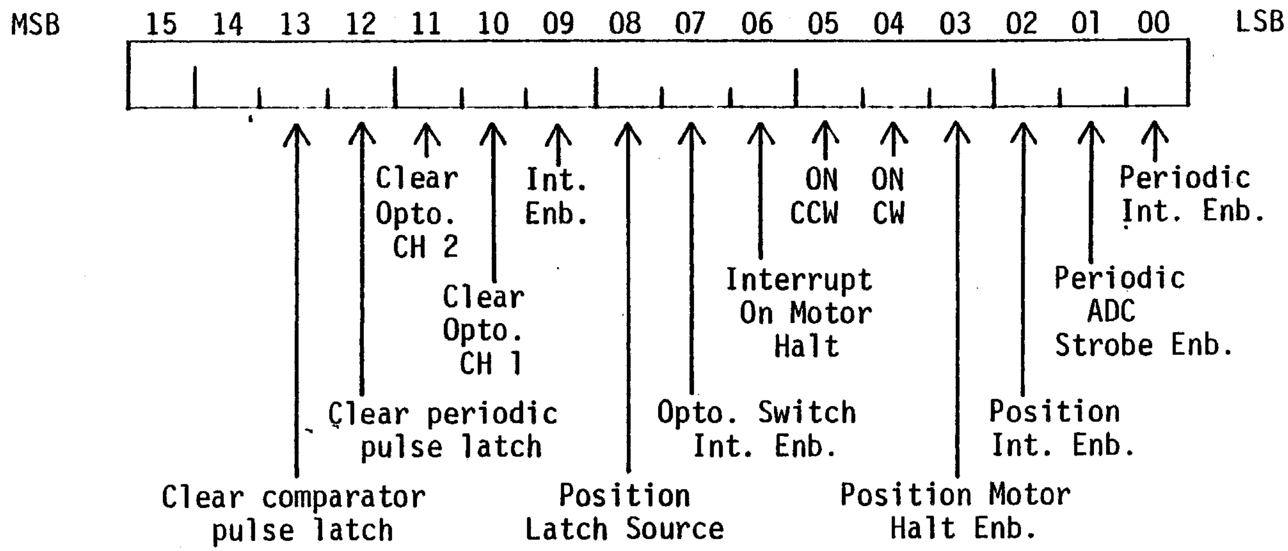


FIGURE 1

Ziptrack Cart Position and Control Board

Function Control Register

Address Offset 16 Octal

N register. (When the divide by N register is loaded by the computer, the number being loaded is simultaneously loaded into the divide by N counter.) Thus, the interrupt occurs in intervals which are periodic in the distance of the sensor cart's motion along its support beam. More directly, it is periodic in the angle through which the motor and the shaft encoder have traveled.

The occurrence of either a periodic interrupt or a periodic ADC strobe is always accompanied by the setting of bit 11 of offset 01, which is the pulse latch of the periodic pulse generator. Any overflow or underflow of the divide by N counter will set this bit, regardless of whether or not the interrupt or ADC strobe functions are enabled.

PERIODIC ADC STROBE FUNCTION

BIT 01: A 1 in this bit enables periodic strobing of the Analog to Digital Converter (ADC). Address offset 14 octal must also be loaded with the number N described below.

The same divide by N counter which handles the periodic interrupt function also handles the periodic ADC strobe function. This function may be selected in addition to or instead of the periodic interrupt function. The periodic ADC strobe shares the same value of N used for the periodic interrupt function. So the two functions cannot be operated independantly at the same time. When the function is selected, a TTL level low true pulse goes out the ribbon connector of the controller and into the ribbon connector of the ADC to give the strobe.

The occurrence of either a periodic ADC strobe or a periodic interrupt is always accompanied by the setting of bit 11 of offset 01, which is the pulse latch of the periodic pulse generator. Any overflow or underflow of the divide by N counter will set this bit, regardless of whether or not the interrupt or ADC strobe functions are enabled.

THE POSITION COMPARATOR INTERRUPT FUNCTION

BIT 02: Loading a 1 into this bit and setting bit 09 enables the position comparator interrupt function. Address offsets 10 and 13 octal must also be loaded, respectively, with the low order word and high order byte of the cart position at which the interrupt is to occur.

The position comparator interrupt is based upon the counter which always contains the current net position in shaft encoder ticks. When the value of this count equals the program controled 24-bit value stored in address offset 10 and 13, the interrupt occurs.

The occurrence of either a comparator interrupt or a comparator motor halt is always accompanied by the automatic setting of bit 10 of offset 01, which is the pulse latch of the comparator. Any position comparator activation, regardless of whether it is programmed to generate an interrupt or motor halt, will set this pulse latch bit.

THE POSITION COMPARATOR MOTOR HALT FUNCTION

BIT 03: Setting this bit enables the position comparator to clear bits 04 and 05 upon detecting an equivalence of the comparator inputs. This causes the motor to turn off when it reaches a programmable position. This function shares the same 24-bit register and comparator as is used by the position comparator interrupt function. The two functions thus cannot be used independantly at the same time. The comparator motor halt function also requires a low order word and high order byte to be loaded at address offsets 10 and 13, respectively.

The occurrence of either a comparator motor halt or a comparator interrupt is always accompanied by the automatic setting of bit 10 of offset 01, which is the pulse latch of the comparator. Any position comparator activation, regardless of whether it is programmed to generate an interrupt or motor halt, will set this pulse latch bit.

TURNING THE MOTOR ON AND OFF

BIT 04: Setting this bit turns the motor on in the clockwise direction, provided that no motor-halting conditions exist and provided that bit 05 is 0.

BIT 05: Setting this bit turns the motor on in the counterclockwise direction, provided that no motor-halting conditions exist and provided that bit 04 is 0.

Note that a delay of about 10 microseconds occurs between the generation of a motor off condition in bits 04 and 05 and the actual gating of this signal onto the cable leading to the motor. This allows for activation of the brake to occur about 10 microseconds before switching off the motor. This in turn reduces inductive voltages at the motor's power relays.

INTERUPT ON MOTOR HALT

BIT 06: A 1 in this bit causes the controler to interrupt when the motor turns off for any reason whatsoever. If this function is enabled, then either a comparator-based stop, a limit-switch stop, or a command stop will generate the interrupt. Note that it is the transition to the off statn the off state itself which generates the interrupt. The interrupt only occurs if the exclusive OR of the last value loaded into (but not nececessarily read from) the two motor on bits is a "1."

OPTICAL SWITCH INTERRUPT ENABLE

BIT 07: A 1 in this bit causes an interrupt to occur if either optical switch channel has recieved a pulse since the last time its pulse latch was cleared. If interrupt capability is re-enabled while bit 7 is still on and either optical switch pulse latch is still "1," then the interrupt will simply be repeated even if no new optical switch pulse has occurred.

POSITION LATCH SOURCE

BIT 08: A 1 in this bit causes the 24-bit position value at address offset 02 and 05 toto be latched only by the occurrence of a pulse on either channel of optical switches. The counter whose output value is frozen, however, continues incrementing or decrementing internally, so that no position information is lost. The value will not latch again until after bit 08 has been cleared or the activated optical switch channel is cleared. (The latter is done by writing a one to the appropriate bit, either 10 or 11, associated with the active switch channel and described below.)

Note that when operating in this mode, the position value read back after an optical switch pulse is free of counter transition errors and hence can be read back a single time without using a software selection routine.

A 0 in this bit leaves position readout unaffected by the occurrence of optical switch values. Yet with a 0 in this bit, a single position readout may contain the effects of counter transitions, necessitating the use of a software selection routine. A working selection routine appears on page 9.

RESETING THE OPTICAL SWITCH CHANNELS

BIT 10: Writing a 1 into this bit clears the channel 1 optical switch, thus leaving a 0 in bit 12 of address offset 01. This channel will then remain inactive until a new low to high transition occurs on its input, even if the input level is high during the clear operation. Writing a 0 to this bit does nothing.

BIT 11: Writing a 1 into this bit clears the channel 2 optical switch, thus leaving a 0 in bit 13 of address offset 01. This channel will then remain inactive until a new low to high transition occurs on its input, even if the input level is high during the clear operation. Writing a 0 to this bit does nothing.

RESETTING THE PERIODIC PULSE LATCH BIT

BIT 12: Writing a 1 into this bit clears the pulse latch of the periodic pulse generator used for periodic interrupts and ADC strobes, thus leaving a 0 in bit 11 of address offset 01. Note that the periodic functions can continue to iterate with or without clearing the pulse latch, as the pulse latch exists for informational purposes only. Writing a 0 into this bit does nothing.

RESETTING THE COMPARATOR PULSE LATCH BIT

BIT 13: Writing a 1 into this bit clears the pulse latch of the comparator pulse generator used for the comparator interrupt and the comparator motor halt, thus leaving a 0 in bit 10 of address offset 01. Note that the comparator functions may continue to operate with or without clearing this pulse latch, as the pulse latch exists for informational purposes only. Writing a 0 into this bit does nothing.

STATUS READBACK BYTE
ADDRESS OFFSET 00 OCTAL

See figure 2 for a diagrammatic summary of the status readback byte.

MOTOR STATUS

BIT 08: A 1 in this bit indicates the controller has received a command to turn on the motor in the clockwise direction and that this command has not yet been cancelled by either a limit switch motor halt or a comparator motor halt. There is, however, one condition in which a 1 in this bit does not indicate that the motor is on. That condition arises when both bit 08 and 09 are on, which is the forbidden condition of turning the motor on in both the clockwise and counterclockwise directions at once. The controller module automatically gates an off condition to the motor when this condition arises, even though bits 08 and 09 remain 1.

BIT 09: A 1 in this bit indicates the controller has recieved a command to turn on the motor in the counterclockwise direction and that this command has not yet been cancelled by either a limit switch motor halt or a comparator motor halt. There is, however, one condition in which a 1 in this bit does not indicate that the motor is on. That condition arises when both bit 08 and 09 are on, which is the forbidden condition of turning the motor on in both the clockwise and counterclockwise directions at once. The controller module automatically gates an off condition to the motor when this condition arises, even though bits 08 and 09 remain 1.

COMPARATOR PULSE LATCH BIT

BIT 10: A 1 in this bit indicates that the position comparator has detected an equivalence condition since the last time this bit was cleared by writing a 1 to bit 13 of offset 16. When either the comparator interrupt or the comparator motor halt functions are enabled, this comparator pulse latch bit is useful in determining what it was that caused the interrupt or motor halt.

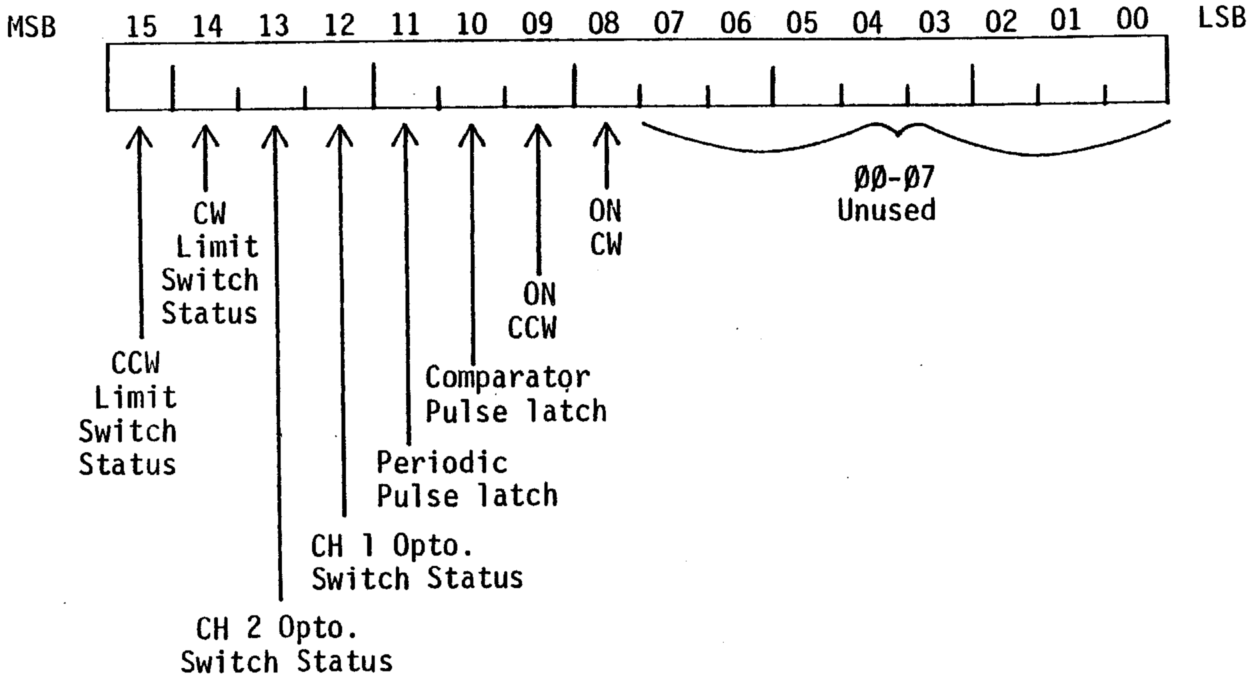


FIGURE 2

Ziptrack Cart Position and Control Board

Status Readback

Address Offset ØØ

PERIODIC PULSE LATCH BIT

BIT 11: A 1 in this bit indicates that a divide by N counter overflow has occurred since the last time this bit was cleared. When either the periodic interrupt or the periodic ADC strobe functions are enabled, this bit is useful in determining what it was that caused the interrupt or ADC strobe.

OPTICAL SWITCH STATUS

BIT 12: A 1 in this bit means that optical switch channel 1 has recieved a pulse since the last time it was cleared by software writing a 1 to bit 10 of address offset 16 octal.

BIT 13: A 1 in this bit means that optical switch channel 2 has recieved a pulse since the last time it was cleared by software writing a 1 to bit 11 of address offset 16 octal.

LIMIT SWITCH STATUS

BIT 14: A 1 in this bit indicates that the clockwise motion limit switch has been activated, thus clearing bit 04 of address offset 16 octal, the clockwise ON command bit. The 1 in this limit switch bit can only be cleared by deactivating the limit switch itself. Thus, no clockwise ON commands will be obeyed until the motor retreats into the counterclockwise direction.

BIT 15: A 1 in this bit indicates that the counterclockwise motion limit switch has been activated, thus clearing bit 05 of address offset 16 octal, the counterclockwise ON command bit. The 1 in this limit switch bit can only be cleared by deactivating the limit switch itself. Thus, no counterclockwise ON commands will be obeyed until the motor retreats into the clockwise direction.

THE MOTOR POSITION REGISTER
ADDRESS OFFSET 02 AND 04 OCTAL

Address offset 02 contains the low order word and address offset

04 contains (in its upper 8 bits) the high order byte of the cart position in shaft encoder ticks. Normally, readout from these addresses may contain counter transition effects which must be removed by a selection routine such as the one below. The exception is when one of the optical switch channel is active during a period when the last number loaded into bit 08 of address offset 16 octal is a 1. In the latter circumstance, the readout value contains no counter transition effects and hence, requires no use of a selection routine.

Notice that in the convention I propose for this module, the clockwise shaft-encoder output gets connected to the downward counter input, and the counterclockwise shaft-encoder input gets connected to the upward counter input. Thus, clockwise motion gives a decreasing number in the position register and counterclocwise motion gives an increasing number in that register.

The following section of code is a selection routine which is adapted to a rapidly changing count value. Be sure to set the processor priority to level 7 before starting this routine.

```
START:  MOV     BASE,R4         ;START LOADING LOW WORD ADDRESS INTO R4
        ADD     #2,R4           ;FINISH SAID LOADING.
        MOV     BASE,R3         ;START LOADING HIGH BYTE ADDRESS INTO R3
        ADD     #4,R3           ;FINISH SAID LOADING.


        MOV     (R4),R0         ;GET LOW WORD
        MOV     (R3),R1         ;GET HIGH BYTE
        MOV     (R4),R2         ;GET LOW WORD AGAIN
        MOV     (R3),R3         ;GET HIGH BYTE FOR SECOND AND LAST TIME
        MOV     (R4),R4         ;GET LOW WORD FOR THIRD AND LAST TIME

                                ;SEARCH FOR A PAIR OF EQUALS FROM AMONG THE
                                ;LOW ORDER WORDS FOUND ABOVE. IF SUCH A PAIR
                                ;IS FOUND, THEN BRANCH TO THE INSTRUCTION THAT
                                ;CHOOSES THE FIRST MEMBER OF FIRST MATCHING
                                ;PAIR OBTAINED AND ITS COMPANION BYTE. (A
                                ;WORD'S "COMPANION BYTE" IS THE BYTE READ IN
                                ;IMMIDIATELY FOLLOWING THE READING OF THAT
                                ;WORD.) IF NO SUCH MATCHING PAIR IS FOUND,
                                ;THEN FIRST VALUE OF THE LOW ORDER WORD
                                ;MUST BE VALID, SO CHOOSE IT AND ITS COMPANION
                                ;BYTE. THE WORD AND BYTE CHOSEN BY THE ABOVE
                                ;CONTINGENCIES ARE VALID COUNT VALUES.

        CMP     R0,R2           ;SEE IF THIS IS A PAIR OF EQUALS.
        BEQ     ALLSET          ;IF SO, BRANCH; ELSE TRY AGAIN.
```

```
        CMP     R2,R4           ;NEW TRY: SEE IF THESE ARE EQUALS.
        BNE     ALLSET          ;NOTE THAT THE ABOVE BRANCHES CAUSE THE FIRST
                                ;WORD-BYTE PAIR TO BE CHOSEN BY STOPPING
                                ;THEM FROM BEING OVERWRITTEN BY THE FOLLOWING
                                ;TWO INSTRUCTIONS, WHICH CHOOSE THE SECOND
                                ;WORD-BYTE PAIR.
        MOV     RO,R2           ;CHOOSE THE SECOND WORD-BYTE COMPANION PAIR.
        MOV     R3,R1           ;FINISH CHOOSING THE SECOND PAIR.


                                ;RO NOW CONTAINS THE VALID LOW-ORDER WORD
                                ;R1 NOW CONTAINS THE VALID HIGH-ORDER BYTE.

ALLSET:                         ;PROCEDE WITH USER PROGRAM
```

THE POSITION COMPARATOR REGISTER
ADDRESS OFFSET 10 AND 12 OCTAL

The contents of this 24-bit register compared with the contents of the 24-bit position counter each time the shaft encoder pulse has finished. When this comparison reveals two identical 24-bit numbers, a signal is formed to activate the position comparator interrupt function and/or the position comparator motor halt function, depending upon the contents of bits 02 and 03 of address offset 16 octal. The count value being compared is not the latched value seen by the computer and sometimes frozen by optical switch functions, but rather the actual, unfrozen current position count.

The low order word of the comparator register is at address offset 10 octal, and the high byte is in the upper 8 bits of address offset 12 octal.

THE PERIODIC FUNCTION DIVIDE BY N REGISTER
ADDRESS OFFSET 14 OCTAL

This register contains the number shaft encoder ticks between interrupts and/or ADC strobes, as controled by bits 00 and 01 of address offset 16 octal. When the cart is moving in the clockwise direction, this number must be stored in high true form. When the cart is moving in the counterclockwise direction, the number must be stored in low true form. That is, for clockwise motion, load the number N; for counterclockwise motion, load the 1's compliment of N (COM N). This means that a complimentary value of N must be loaded each time the cart changes directions if the same periodicity is to be used.

Note that the input cables must be configured so that clockwise motion yeilds downward counting and counterclockwise motion yeilds upward counting in order for the module to function as described above. If the input cables for clockwise and counterclockwise shaft-encoder pulses are reversed, then it is the the clockwise motion that requires the 1's compliment of N.

THE COUNTER CLEAR FUNCTION
ADDRESS OFFSET 06 OCTAL

Addressing addressing address offset 06 octal in any way causes all counters on the board to be reset to 0. This includes both the position counter and the counter contained in the divide by N circuitry. The divide by N register itself is not affected, but its value must be reloaded anyway to insure consistant operation. For instance, with counterclockwise motion, clearing the counters without reloading the divide by N register causes divide by the compliment of N for the first cycle. With clockwise motion, failure to reload the divide by N register after a counter clear causes divide by 1 for the first cycle. None of the other functions require register re-loading after a counter clear.

PROJECT: GENERAL HARDWARE PROJECTS

DOC. TYPE: TECHNICAL NOTE
TITLE: Ziptrack Manipulator Stepping Motor Controller Documentation

SUBJECT:

GROUP: PDP-11 HARDWARE
AUTHOR: Aaron Lynch

DOC NO.: 42
DATE: 20-JUL-81 09:35:58
FILE: MANIP.RNO

BASE ADD: 1641008

1.0 INTRODUCTION

The Ziptrack Stepping Motor Controller is the source of the pulse trains that determine the speed, distance, and direction for which the stepping motor drivers (translators) run the motors themselves. It does this for a maximum of two motors running at once, but the outputs are multiplexed so as to run a maximum total of four motors if there are always at least two idle.

Nine registers control all the programable functions of the controller. These nine registers are the control and status register (CSR), the channel 1 rate divisor word, the channel 1 rate divisor byte, the channel 1 pulses-remaining word, the channel 2 rate divisor word and byte, the channel 2 pulses-remaining word, the pulses-remaining comparator word, and the limit switch status word. All of these registers are of the read/write type, except for the limit switch status word, which is of read only type.

The first of these words, the control and status word, occurs at address offset 00 octal. Since the bits of this word perform a variety of functions, I will give a bit by bit description of the CSR. See figure 1 for a diagrammatic summary of the CSR.

BIT 00: Setting this bit causes the channel 1 pulse train to start if the channel 1 effective rate and pulses-remaining are not zero and if no channel 1 limit switch is on.

BIT 01: Setting this bit causes the channel 2 pulse train to start if the channel 2 effectivee rate and pulses-remaining are not zero and if no channel 2 limit switch is on.

BITS 02 THROUGH 07: These bits comprise the centralized rate-multiplier factor. This factor scales the frequeencies of the two channels of pulses by the same proportion. The inputs of the channel 1 and channel 2 rate divisors both come from the output of this single rate multiplier. The multiplier parameter may take any value from 0 to 63 decimal. The parameter's value, divided by 64, is the fraction applied to the frequency of the 3 MHz. clock before going into the separate channels of freequency dividers. The effective frequency of a channel is then equal to 3 MHz. times the multiplier parameter divided by 64 and divided by the value of the channel's rate divisor. The basic purpose of the rate divisor is to provide the capability of acceleration by proportions that are uniform across both channels, thus preserving any desired geometric ratios

BIT 8: This bit simultaneously determines the direction of rotation of both running motors (or of one motor if only one is enabled.) A 1 in this bit specifies counterclockwise rotation, and a 0

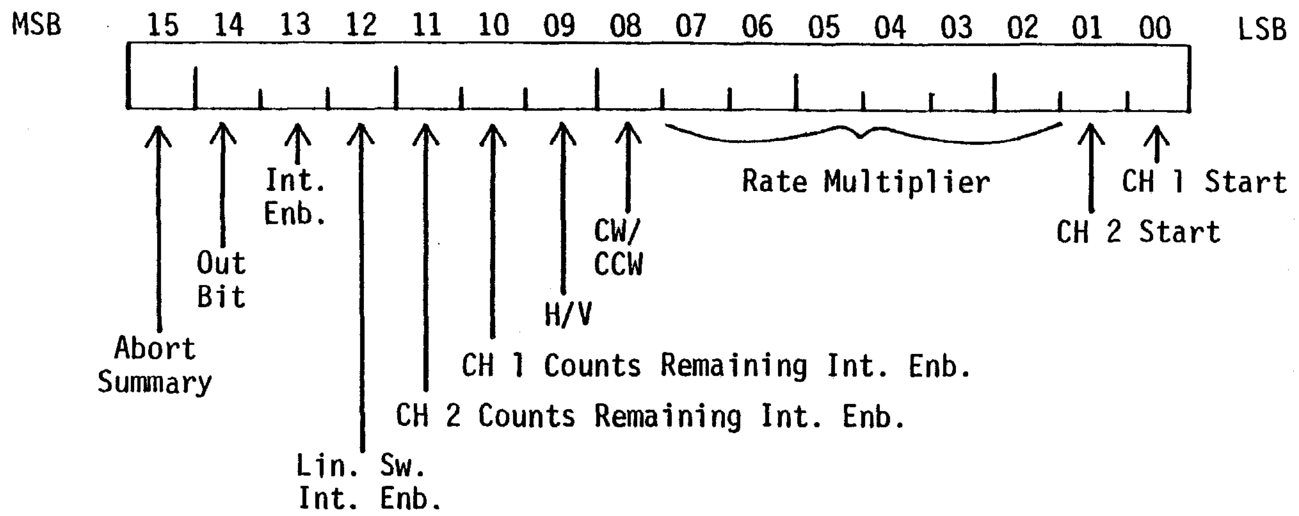


FIGURE 1

Ziptrack Stepping Motor Controller

Control and Status Register

Address Offset ØØ

specifies clockwise rotation.

BIT 9: This bit determines whether the two channels of pulses are going to the horizontal motor pair or the vertical motor pair. A 1 in this bit specifies vertical, and a 0 specifies horizontal.

BIT 10: This bit in conjunction with bit 13 (master interrupt enable) will cause an interrupt to occur if the channel 1 counts-remaining word becomes equal to the counts-remaining comparator word.

BIT 11: This bit in conjunction with bit 13 (master interrupt enable) will cause an interrupt to occur if the channel 2 counts-remaining word becomes equal to the counts-remaining comparator word.

Note that the counts-remaining comparator word serves both channels at once. Thus, if both bits 10 and 11 are set, the interrupt will occur when either of the two channel's counts-remaining words reaches the value of the counts-remaining comparator word. If neither bit is set, then equivalence of the above words has no external effect.

BIT 12: This bit in conjunction with bit 13 causes an interrupt to occur whenever any one of the limit switches is activated.

BIT 13: This bit is the master interrupt enable for the controller module. There must be a 1 in this bit in order for any of the above interrupts to occur. Also, this bit is automatically cleared during the start of the interrupt itself, so that the software must reload a 1 into this bit in order to enable a subsequent interrupt. All of the interrupts of the controller module carry the same interrupt vector.

BIT 14: This bit is output through connector pin 40 with ground on pin 39. It is a general purpose programmable output bit. In the present ziptrack, it is used to clamp the integrators after each run of the cart.

BIT 15: This bit contains the limit switch (and hence, the abort) summary. Whenever any limit switch is thrown even momentarily, this bit becomes a 1 and all motors are stopped. Motion cannot be resumed until the bit is cleared by software. Clearing the bit is done by writing a 1 into bit 15 of address offset 20 octal. Bit 15 cannot be modified by writing to it directly.

I will now describe the functions of the words at other address offsets.

OFFSET 02: This register contains the low order word of the rate (frequency) divisor for channel 1. This word can be loaded without restriction even during motor motion provided that its companion high order byte (address offset 04) is zero and is to remain zero in the new 24 bit rate divisor.

OFFSET 04: This register contains the high order byte of the channel 1 rate divisor. Normally, this register remains zero unless the motor run reaches its top speed at less than 45 Hz. Never load this byte during motor motion without also loading the low order word in an adjacent instruction. Also, bits 00 and 02 must be cleared momentarily (for a few microsecends) while the word and byte are being loaded. The program which performes this word-byte load must not be interrupted between the two move instructions that accomplish the load. This word and its companion byte can be loaded without restriction during motor off condition.

The actual frequency output on channel 1 is equal to the input frequency of the channel 1 divisor circuit divided by the sum of its 24-bit register and the number 1. Thus, if the above word and byte form the number N, then the frequency is divided by N+1. If the rate multiplier (CSR bits 02 through 07) contains the number M, then this frequency (F) is given by: $F = 46875M/(N+1)$.

OFFSET 06: This word contains the 16-bit number of pulses remaining to be given to the channel 1 motor. Once this number is loaded and the motor turned on, the number automatically decriments once for each pulse issued to the motor. The number is thus modified by both the program and by the normal internal operation of the controller module. When the number reaches 0, the controller automatically shuts off the motor, (ceases to issue pulses to its translator/driver).

OFFSET 10: This register contains the low order word of the channel 2 rate divisor. It functions in the same way and with the same restrictions as the channel 1 rate divisor low word. Its value does not, however, depend on the value of the channel 1 word.

OFFSET 12: This register contains the high order byte of the channel 2 rate divisor. Again, it performs the same value as the offset 04 register, only with an independant value.

OFFSET 14: This register contains the pulses-remaining for channel 2. It works the same way as the counts-remaining register for channel 1.

OFFSET 16: This register contains the counts-remaining comparator word. The role of this word is discribed under the function of CSR bits 10 and 11 above.

OFFSET 20: Readout of this register gives the status of all limit switches in bits 0 through 8. Bit 15 contains the limit switch summary, a duplicate of CSR bit 15. The only difference is that writing a 1 to bit 15 of this register clears bit 15 of both registers. See figure 2 for a diagrammatic summary of the limit switch status word.

When the content of this register makes a transition from zero to any other value, it clears CSR bits 00 and 01, thus stopping all motor motion. These bits can then be reloaded only by the program. Any other transitions in the limit switch status register have no effect upon the CSR bits. Thus, if the program reloads those CSR bits and a new limit switch is triggered without the one that is already on being turned off, the CSR will not be automatically changed and each motor will not necessarily be halted.

What the limit switches always do is to inhibit all further motion in the direction of the activated switch itself. Thus, if the channel 1 horizontal clockwise limit switch is thrown, the channel 1 horizontal motor can move only in the counterclockwise direction until the limit switch de-activates. Software cannot affect this restriction.

One limit switch, whose status appears in bit 8, is the unforgiving motor abort. This switch clears CSR bits 00 and 01 and does not allow them to be reloaded until after someone goes out and manually moves the manipulator or else somehow changes the signal level coming from this limit swich. This switch should be used for sensing contact between the cart's track and the magnet so that the track is not too badly damaged by the collisions which cause such contact.

The following is a list of the limit switch states appearing in bits 00 through 07.

BIT 00	Channel 1	Horizontal	Clockwise	Limit Switch
BIT 01	Channel 1	Horizontal	Counterclockwise	Limit Switch
BIT 02	Channel 1	Vertical	Clockwise	Limit Switch
BIT 03	Channel 1	Vertical	Counterclockwise	Limit Switch
BIT 04	Channel 2	Horizontal	Clockwise	Limit Switch
BIT 05	Channel 2	Horizontal	Counterclockwise	Limit Switch
BIT 06	Channel 2	Vertical	Clockwise	Limit Switch
BIT 07	Channel 2	Vertical	Counterclockwise	Limit Switch

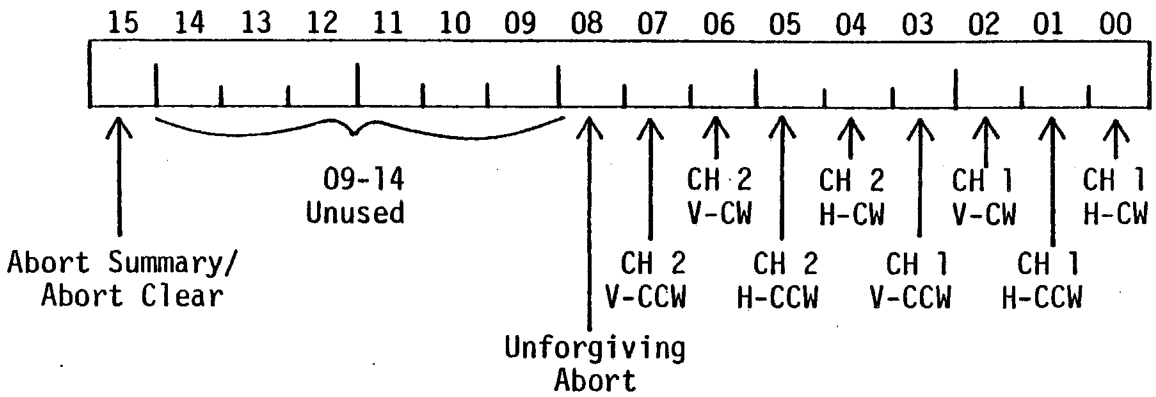


FIGURE 2

Ziptrack Stepping Motor Controller

Abort Status and Clear Word

Address Offset 2Ø

PROJECT:	GENERAL HARDWARE PROJECTS
DOC. TYPE:	TECHNICAL NOTE
TITLE:	Ziptrack Manipulator Position Monitor Documentation
SUBJECT:	
GROUP:	PDP-11 HARDWARE
AUTHOR:	Aaron Lynch
DOC NO.:	41
DATE:	17-JUL-81 09:58:10
FILE:	POSIT.RNO
BASE ADD:	1640008

1.0 INTRODUCTION

The Ziptrack Manipulator Position Monitor module serves the sole purpose of counting the output pulses from the manipulator shaft encoders and making the counts available to the computer. The count value it provides is the net number of pulses recieved from each shaft encoder. That is, it always gives the total number of clockwise pulses recieved minus the total number of counterclockwise pulses recieved. It does this for each of a maximum of five independant shaft encoders.

Each shaft encoder channel is counted to an accuracy of 24 bits. Reading out the count therefore requires reading a word and a byte for each channel. The word contains the low order 16 bits and the byte contains the high 8 bits.

The module was designed with the intention of using a special software data selection routine to ensure that the software ignores any count values that were taken while the counters had not yet settled down from their last shaft encoder pulse. This routine simplified the hardware design, but it constitutes an unconventional method of data readout. That routine is therefore provided as part of this document.

The selection routine is based on the fact that three consecutive readouts of a single channel's count can contain at most one count value from the counter's unsettled state. That this is so follows from the fact that the cycle time of the shaft encoder pulses is at least 20 times slower than the instruction cycle of the slowest PDP-11 computer. So what the selection routine does is take three consecutive readouts. If their values are all different, then the second (middle) readout was unsettled. This means that the first readout was correct, and the routine therefore choses the first one in this case. If two or more readouts are identical, then each of the identical readouts is correct and the routine chooses the first readout it finds existing in such an identical pair.

The following is a list of the offset address, the transaction type, and the function of each word and byte in the position monitor. Note that although the high order byte of each channel is read in from an even address, it will be placed in the upper 8 bits of the word into which it is being read.

00	read only	channel 1 low order word
02	read only	channel 2 low order word
04	read only	channel 3 low order word
06	read only	channel 4 low order word
10	read only	channel 5 low order word

12		read only	channel 1 high order byte
14		read only	channel 2 high order byte
16		read only	channel 3 high order byte
20		read only	channel 4 high order byte
22		read only	channel 5 high order byte
24	25	read/write	channel 1 counter clear
26	27	read/write	channel 2 counter clear
30	31	read/write	channel 3 counter clear
32	33	read/write	channel 4 counter clear
34	35	read/write	channel 5 counter clear
36	37	read/write	all channel counter clear

Note that the counter clearing functions operate simply by addressing the specified locations. They will operate regardless of what data values the program specifies in a write transaction. A read at those addresses will always return a 0 to the computer.

THE SINGLE CHANNEL DATA SELECTION SOFTWARE

The following is a data selection routine for a single counter channel. Assume that the address of the low order word of the 24-bit counter value is labeled BASE. BASE+12(octal) is then the address of the high order byte.

The routine is adapted to a rapidly changing count value. Therefore, be sure to set the processor priority to level 7 before starting this routine. This will prevent the routine from being interrupted, thus assuring the integrity of the selected data.

```
START:  MOV     BASE,R4         ;START LOADING LOW WORD ADDRESS INTO R4
        ADD     #2,R4           ;FINISH SAID LOADING.
        MOV     BASE,R3         ;START LOADING HIGH BYTE ADDRESS INTO R3
        ADD     #12,R3          ;FINISH SAID LOADING.


        MOV     (R4),RO         ;GET LOW WORD
        MOV     (R3),R1         ;GET HIGH BYTE
        MOV     (R4),R2         ;GET LOW WORD AGAIN
        MOV     (R3),R3         ;GET HIGH BYTE FOR SECOND AND LAST TIME
        MOV     (R4),R4         ;GET LOW WORD FOR THIRD AND LAST TIME
```

```
                                  ;SEARCH FOR A PAIR OF EQUALS FROM AMONG THE
                                  ;LOW ORDER WORDS FOUND ABOVE. IF SUCH A PAIR
                                  ;IS FOUND, THEN BRANCH TO THE INSTRUCTION THAT
                                  ;CHOOSES THE FIRST MEMBER OF FIRST MATCHING
                                  ;PAIR OBTAINED AND ITS COMPANION BYTE. (A
                                  ;WORD'S "COMPANION BYTE" IS THE BYTE READ IN
                                  ;IMMIDIATELY FOLLOWING THE READING OF THAT
                                  ;WORD.) IF NO SUCH MATCHING PAIR IS FOUND,
                                  ;THEN FIRST VALUE OF THE LOW ORDER WORD
                                  ;MUST BE VALID, SO CHOOSE IT AND ITS COMPANION
                                  ;BYTE. THE WORD AND BYTE CHOSEN BY THE ABOVE
                                  ;CONTINGENCIES ARE VALID COUNT VALUES.

        CMP     RO,R2             ;SEE IF THIS IS A PAIR OF EQUALS.
        BEQ     ALLSET            ;IF SO, BRANCH; ELSE TRY AGAIN.
        CMP     R2,R4             ;NEW TRY: SEE IF THESE ARE EQUALS.
        BNE     ALLSET            ;NOTE THAT THE ABOVE BRANCHES CAUSE THE FIRST
                                  ;WORD-BYTE PAIR TO BE CHOSEN BY STOPPING
                                  ;THEM FROM BEING OVERWRITTEN BY THE FOLLOWING
                                  ;TWO INSTRUCTIONS, WHICH CHOOSE THE SECOND
                                  ;WORD-BYTE PAIR.
        MOV     RO,R2             ;CHOOSE THE SECOND WORD-BYTE COMPANION PAIR.
        MOV     R3,R1             ;FINISH CHOOSING THE SECOND PAIR.

                                  ;RO NOW CONTAINS THE VALID LOW-ORDER WORD.
                                  ;R1 NOW CONTAINS THE VALID HIGH-ORDER BYTE
                                  ;IN ITS UPPER 8 BITS.

ALLSET:                           ;PROCEDE WITH USER PROGRAM
```

Ziptrack Tape Dump Procedure

10/1/85

1. Remove yellow write ring from Ziptrack tape.

2. Label tape as "NO VAULT" and with its name eg ZJ001 where ZJ is required and 001 arbitrary.

3. Submit tape to operator on 7th floor of Computing Center.

4. Log in to your account on CYBER:If the terminal is on 4800 baud hit the space bar, and wait for the "FERMILAB=" and type 4 and carriage return. When you are logged on, if the terminal does not echo type: esc then E then P then = then Y and carriage return.

5. Make a copy of Willy Yang's file ZIPDMP. Call your copy ZIPDMP:

   ```
   GET, ZIPDMP/un=92836
   SAVE(ZIPDMP=ZIPDMP)
   ```

6. In your copy of ZIPDMP you need to change Willy Yang's password, username, charge and tape name to your own using:

   ```
   ICE,ZIPDMP/GET
   s/Willy Yang's username/your username/1:22
   s/Willy Yang's charge/your charge/1:22
   s/Willy Yang's password/your password/1:22
   s/Willy Yang's tapename/your tapename/1:22
   ER
   ```

7. Submit the file ZIPDMP: SUBMIT,ZIPDMP,B

8. Check the status of the job by typing: STATUS,JSN and looking for the job name. When the job is no longer present, that means it has finished executing. Go pick up the output.

9. When you need the tape ask the operator or wait ~24 hours for it to be placed on the tape rack.

10. Only problem could be with your PROCFIL.